

AD-A087 186

ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG--ETC F/G 1/5
PAVEMENT EVALUATION AND OVERLAY DESIGN USING VIBRATORY NONDESTR--ETC(U)
MAR 80 R A WEISS DOT-FA73WAI-377

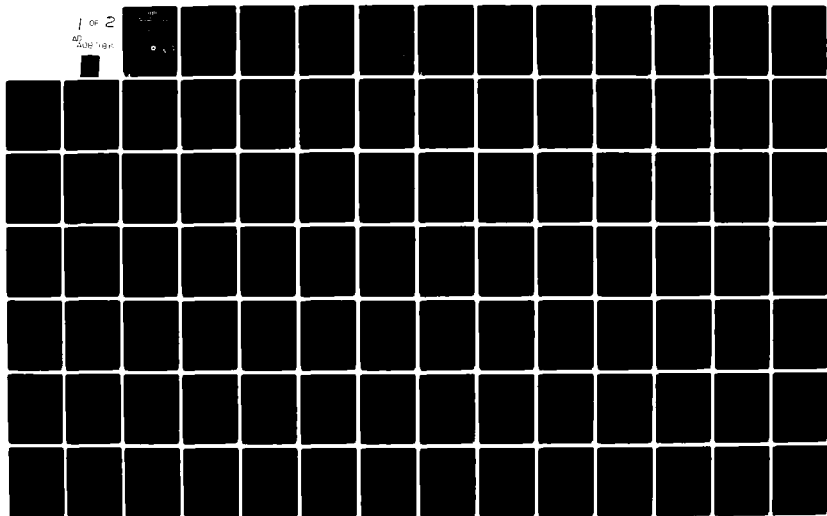
UNCLASSIFIED

FAA-RD-77-186-VOL-1

NL

1 of 2

AD-A087 186



Report No. FAA-RD-77-186-I

LEVEL

(12)

**PAVEMENT EVALUATION AND OVERLAY DESIGN USING
VIBRATORY NONDESTRUCTIVE TESTING AND
LAYERED ELASTIC THEORY**

Volume I

Development of Procedure

Richard A. Weiss

**U. S. Army Engineer Waterways Experiment Station
Geotechnical Laboratory
P. O. Box 631, Vicksburg, Miss. 39180**



**MARCH 1980
FINAL REPORT**

Document is available to the public through the
National Technical Information Service,
Springfield, Va. 22151

Prepared for

**U. S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D. C. 20591**

**DTIC
ELECTE
JUL 25 1980**

ADA087186

DDC FILE COPY

80 7 24 015

NOTICES

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

19) 77-186-VOL-1

Technical Report Documentation Page

1. Report No. 18) FAA-RD-77-186-I	2. Government Accession No. AD-A087186	3. Recipient's Catalog No.
4. Title and Subtitle 6) PAVEMENT EVALUATION AND OVERLAY DESIGN USING VIBRATORY NONDESTRUCTIVE TESTING AND LAYERED ELASTIC THEORY. Volume I: DEVELOPMENT OF PROCEDURE	5. Report Date March 1980 411712	6. Performing Organization Code
7. Author(s) 10) Richard A. Weiss	8. Performing Organization Report No. 11) Mar 80 2	10. Work Unit No. (TRAIS)
9. Performing Organization Name and Address U. S. Army Engineer Waterways Experiment Station Geotechnical Laboratory P. O. Box 631, Vicksburg, Miss. 39180	11. Contract or Grant No. 15) DOT-FA73WAI-377	13. Type of Report and Period Covered Final Report Oct 1977 to Dec 1978
12. Sponsoring Agency Name and Address U. S. Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, D. C. 20591	14. Sponsoring Agency Code	
15. Supplementary Notes		
16. Abstract A procedure is developed for determining the allowable load-carrying capacities and the required overlay thicknesses of airport pavements. A layered elastic theory approach is used with vibratory nondestructive tests supplying the dynamic responses of pavements. For a given pavement, a computer program SUBE is used to determine the value of the subgrade Young's modulus from the measured dynamic responses, and a computer program PAVEVAL, which is based on the layered elastic theory, is used to calculate the allowable load-carrying capacity and the required overlay thickness. Limiting subgrade strains and horizontal stresses in pavement layers are used as criteria for determining load-carrying capacities and overlay thickness requirements. Single- and multiple-wheel loadings are considered. Volume II of this report presents a validation of these procedures for three airport sites.		
17. Key Words Nondestructive testing Layered elastic theory Pavement design and evaluation	18. Distribution Statement Document is available to the public through the National Technical Information Service, Springfield, Va. 22151	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 174
		22. Price

411412 xll

METRIC CONVERSION FACTORS

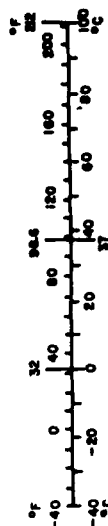
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
AREA				
sq in	square inches	6.5	square centimeters	cm ²
sq ft	square feet	0.09	square meters	m ²
sq yd	square yards	0.8	square meters	m ²
sq mi	square miles	2.6	square kilometers	km ²
acres	acres	0.4	hectares	ha
MASS (weight)				
oz	ounces	28	grams	g
lb	pounds	0.46	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
VOLUME				
teaspoon	teaspoons	5	milliliters	ml
Tablespoon	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cup	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.96	liters	l
gal	gallons	3.8	liters	l
cu ft	cubic feet	0.03	cubic meters	m ³
cu yd	cubic yards	0.76	cubic meters	m ³
TEMPERATURE (exact)				
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

* 1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weight and Measures, Price \$2.25, SD Catalog No. C13.10-286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
km	kilometers	1.1	miles	mi
		0.6	miles	mi
AREA				
cm ²	square centimeters	0.16	square inches	in ²
m ²	square meters	1.2	square yards	yd ²
km ²	square kilometers	0.4	square miles	mi ²
ha	hectares (10,000 m ²)	2.5	acres	ac
MASS (weight)				
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
VOLUME				
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m ³	cubic meters	36	cubic feet	ft ³
m ³	cubic meters	1.3	cubic yards	yd ³
TEMPERATURE (exact)				
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



PREFACE

This study was conducted during the period October 1977 to December 1978 by personnel of the Geotechnical Laboratory (GL), U. S. Army Engineer Waterways Experiment Station (WES), for the U. S. Department of Transportation, Federal Aviation Administration, as a part of Inter-Agency Agreement No. DOT FA73WAI-377, "New Pavement Design Methodology." ✓

The study was conducted under the general supervision of Messrs. J. P. Sale and R. G. Ahlvin, Chief and Assistant Chief, respectively, of GL; R. L. Hutchinson and H. H. Ulery, Jr., Chief and Principal Technical Advisor, respectively, of the Pavement Systems Division; and under the direct supervision of A. H. Joseph, Chief of the Engineering Investigation Testing and Validation Group; and J. W. Hall, Jr., Chief of the Prototype Testing and Evaluation Unit. The programming for this study was accomplished in part by Mr. Ricky Austin, Research and Analysis Group. Significant contributions were made by Messrs. J. L. Green and A. J. Bush III of the Prototype Testing and Evaluation Unit, and by Dr. W. R. Barker of the Research and Analysis Group. The report was written by Dr. R. A. Weiss.

COL John L. Cannon, CE, and COL Nelson P. Conover, CE, were Directors of the WES during the conduct of this study and the preparation of this report. The Technical Director was Mr. F. R. Brown.

Accession For	
NTIS GRA&I	
DDC TAB	
Unannounced	
Justification	
By	
Distribution/	
Availability Codes	
Dist	Available and/or special
A	

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
BACKGROUND	1
OBJECTIVES	4
SCOPE	4
MATERIAL PARAMETERS AND FAILURE CRITERIA REQUIRED FOR PAVEMENT EVALUATION AND OVERLAY DESIGN	6
GENERAL CONSIDERATIONS	6
FAILURE IN AC PAVEMENTS	7
FAILURE IN PCC PAVEMENTS	7
PAVEMENT EVALUATION AND OVERLAY DESIGN	8
SELECTION OF THE PAVEMENT AND AIRCRAFT PARAMETERS FOR THE COMPUTER PROGRAMS SUBE AND PAVEVAL	10
GENERAL CONSIDERATIONS	10
PAVEMENT LAYER THICKNESSES	10
POISSON'S RATIO	10
YOUNG'S MODULUS	11
FLEXURAL STRENGTH	14
AIRCRAFT CHARACTERISTICS	15
LIMITING STRESS AND STRAIN CONDITIONS	17
AC PAVEMENTS	17
PCC PAVEMENTS	20
SUBGRADE YOUNG'S MODULUS DETERMINED BY VIBRATORY NONDESTRUCTIVE TESTING OF PAVEMENTS	23
GENERAL CONSIDERATIONS	23
VIBRATORY NONDESTRUCTIVE TEST DATA	23
NONLINEAR DYNAMIC THEORY OF PAVEMENT RESPONSE	24
DYNAMIC PAVEMENT RESPONSE COMPUTER PROGRAM SUBE	26
NUMERICAL VALUES OF THE PREDICTED SUBGRADE YOUNG'S MODULUS	26
LAYERED ELASTIC THEORY CALCULATION OF THE STRESS AND THE STRAIN IN PAVEMENTS	34
GENERAL CONSIDERATIONS	34
BISAR COMPUTER PROGRAM	34

	<u>Page</u>
ALLOWABLE LOAD-CARRYING CAPACITY AND REQUIRED OVERLAY THICKNESS OF PAVEMENTS	36
GENERAL CONSIDERATIONS	36
CHOICE OF ELASTIC MODULI FOR PAVEVAL COMPUTER PROGRAM . . .	36
SINGLE-WHEEL LOADING	36
MULTIPLE-WHEEL LOADING	39
ALLOWABLE LOAD-CARRYING AND REQUIRED OVERLAY THICKNESS FOR AC AND PCC PAVEMENTS	42
NUMERICAL VALUES OF THE ALLOWABLE LOAD-CARRYING CAPACITY AND THE REQUIRED OVERLAY THICKNESS	45
SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS	70
SUMMARY	70
CONCLUSIONS	71
RECOMMENDATIONS	71
REFERENCES	74
APPENDIX A: COMPUTER PROGRAM SUBE	A-1
APPENDIX B: COMPUTER PROGRAM PAVEVAL	B-1

INTRODUCTION

BACKGROUND

The increasing expense of pavement construction and rehabilitation makes it essential to have a fast and reliable method of accurately predicting the allowable load-carrying capacity and the required overlay thickness for pavement upgrading. Vibratory nondestructive testing can play an important part for the rapid evaluation of airport pavements.¹⁻⁶ The U. S. Army Engineer Waterways Experiment Station (WES) was requested to develop a pavement evaluation method based on vibratory nondestructive testing combined with layered elastic theory. This study combines the layered elastic theory for calculating stress and strain in a pavement with the nonlinear theory of dynamic pavement response that describes the vibratory nondestructive test data to produce a working method, including computer programs, for evaluating pavements and designing overlays.

The combined method of layered elasticity theory and vibratory nondestructive testing is compared with the conventional method that uses the California Bearing Ratio (CBR) for evaluating asphaltic concrete (AC) pavements and with the Westergaard method for evaluating portland cement concrete (PCC) pavements.⁷ It is also compared with the pavement evaluation method that uses a correlation between the strength of a pavement and the dynamic stiffness modulus (DSM) that is obtained from vibratory nondestructive testing.¹

The CBR and Westergaard methods require destructive tests to measure the CBR and the coefficient of subgrade reaction, respectively. To circumvent the destructive tests, a vibratory nondestructive test method, which directly correlates the allowable load-carrying capacity and the required overlay thickness to a mechanical impedance that is measured at the pavement surface (the DSM), was developed at the WES for evaluating AC and PCC pavements.

The DSM is obtained from vibratory nondestructive test data that are obtained with the WES electrohydraulic vibrator, which can

generate dynamic loads up to 15 kips with a constant 16-kip static load (WES 16-kip vibrator) and a constant frequency of 15 Hz.⁴ These data consist of dynamic load-deflection curves that are measured at the pavement surface. The dynamic load-deflection curves are nonlinear in general, and the DSM is the slope of the dynamic load-deflection curve for a dynamic load of about 10-14 kips. The measured DSM is corrected to a common pavement temperature of 70°F, and the corrected value of the DSM is correlated to the allowable load-carrying capacity and the required overlay thickness of a pavement.^{1,6} The DSM method is empirical and does not take into consideration the layered elastic structure of the pavement or the interface conditions between the pavement layers.

In order to improve on the method of directly correlating pavement performance with vibratory nondestructive test data, an attempt was made to combine the layered elastic theory of pavements with the pavement impedance values measured by vibratory nondestructive tests. In this way, the pavement structure could be considered. The layered elastic model of pavements required the Young's modulus and the Poisson's ratio of the subgrade and pavement layers to be known. The elastic moduli of the pavement layers are estimated by various means, and only the subgrade Young's modulus is obtained by vibratory nondestructive tests.

The pavement evaluation method presented herein consists of determining the subgrade Young's modulus from the dynamic response of a pavement measured by vibratory nondestructive tests and using the determined value of the subgrade Young's modulus in the layered elastic theory to calculate the allowable load-carrying capacity and the required overlay thickness of a pavement. Two computer programs, SUBE and PAVEVAL, are used for the necessary computations and to obtain the results.

The subgrade Young's modulus is determined from dynamic load-deflection curves that are measured at the pavement surface. In general, these dynamic load-deflection curves are nonlinear, and a nonlinear dynamic theory is required to extract the value of the subgrade Young's modulus from these measured curves. The nonlinear

dynamic theory is used to remove the extraneous effects of the static and dynamic loads developed by the vibrator on the predicted values of the subgrade Young's modulus.^{3,4} The value of the subgrade Young's modulus used for calculating the allowable load-carrying capacity and the required overlay thickness of a pavement should reflect only the stress conditions in the sugrade due to the aircraft loading and the natural overburden pressure. The computer program SUBE was developed from the nonlinear theory of pavement response to dynamic loads and is used to determine the subgrade Young's modulus from the measured dynamic load-deflection curves.

Within the context of the layered elastic theory, pavements are represented by a stack of elastic layers, the subgrade being of infinite extent. This layered elastic theoretical model of a pavement structure is used to calculate the elastic stress and strain at any point in the pavement or the subgrade. Each pavement layer is characterized by a Poisson's ratio (ν), a Young's modulus (E), and a layer thickness (h). The Shell BISAR computer program is based on the layered elastic theory and relates the stress and the strain in each pavement layer to the static load applied to the surface of a pavement. The computer program PAVEVAL that is used for pavement evaluation and overlay design is a modification of the BISAR program. Figure 1 represents a typical pavement structure subjected to a loading according to the layered elastic theory approach.

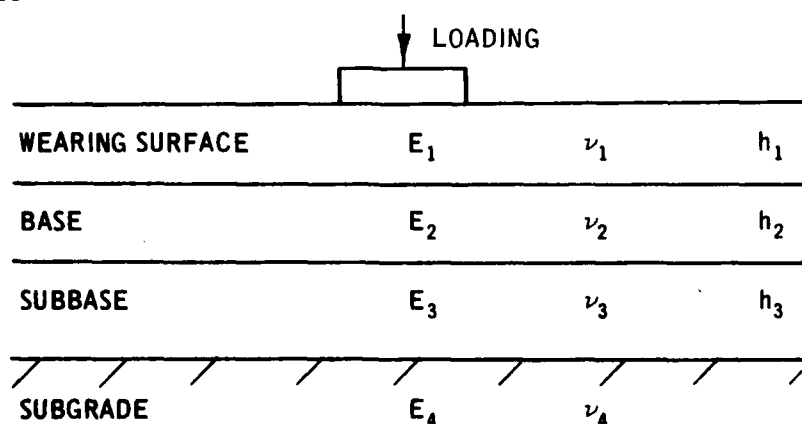


Figure 1. Typical pavement structure with loading according to the layered elastic theory

The computer programs SUBE and PAVEVAL were developed on the IBM 360/65 computer and are designed for practical use by pavement engineers.

OBJECTIVES

A pavement evaluation procedure is required that will use vibratory nondestructive testing and analysis of data to obtain the value of the subgrade modulus for input into the layered elastic theory for calculating stresses and strains in a pavement. The basic objectives of this study are:

- a. To further develop and evaluate a theoretical procedure for determining the subgrade Young's modulus from vibratory nondestructive test data.
- b. To determine the allowable load-carrying capacity for AC and PCC pavements for single- and multiple-wheel loadings using the subgrade modulus in the layered elastic theory.
- c. To determine the overlay thickness required to upgrade AC and PCC pavements for single- and multiple-wheel loadings using elastic pavement parameters calculated from vibratory nondestructive testing results in the layered elastic theory.

SCOPE

To achieve these objectives, theoretical and experimental work was done.

THEORETICAL STUDIES

The theoretical studies included:

- a. A logical method of selecting the values of the elastic moduli of each pavement layer.
- b. The development of the nonlinear dynamic computer program SUBE to predict the values of the subgrade Young's modulus from measured vibratory nondestructive test data.
- c. The determination of the limiting vertical strain in the subgrade and the limiting tensile strain in the AC layer of AC pavements, and a limiting tensile stress criterion in PCC pavements as design criteria to be used with the layered elastic pavement model.
- d. The development of the PAVEVAL layered elastic computer program to calculate the allowable load-carrying capacity and the required overlay thickness for AC and PCC pavements

with single- and multiple-wheel loadings. A comparison with conventional CBR and Westergaard methods is made.

EXPERIMENTAL STUDIES

Dynamic load-deflection curves and CBR values were measured in the field for PCC and AC pavements.

MATERIAL PARAMETERS AND FAILURE CRITERIA REQUIRED FOR PAVEMENT EVALUATION AND OVERLAY DESIGN

GENERAL CONSIDERATIONS

Repeated aircraft loadings on a pavement will eventually lead to a failure of the pavement. The ultimate purpose of the nondestructive testing of a pavement is to estimate the allowable load-carrying capacity of a pavement for a specified number of yearly load repetitions or to determine the overlay thickness required to upgrade a pavement when the operating aircraft weight and yearly number of load repetitions are specified.¹ The estimation of the allowable load-carrying capacity and the required overlay thickness requires a knowledge of the failure processes that occur in AC and PCC pavements. Vibratory nondestructive testing should supply some of the pavement parameters that enter into the physical description of the failure processes.²⁻⁴

Pavements fail for a variety of reasons. Many pavements fail because the pavement does not properly protect the subgrade from large stresses and strains that can cause excessive plastic and elastic deformation of the soil in the subgrade. Experience has shown that the condition of failure in AC pavements may be described by a limiting elastic (resilient) vertical strain in the top of the subgrade and a limiting tensile strain at the bottom of the AC pavement layer, while the condition of failure in PCC pavements can be described by a limiting tensile stress at the bottom of the PCC layer.^{8,9} These limiting values of stress and strain are related to the allowable load-carrying capacity and the required overlay thickness of a pavement through the structure of the pavement, i.e., through the thickness and the material type of each layer of the pavement and the subgrade.

The materials in the pavement layers must be described by material parameters, which determine the stress-strain characteristics. The proper mechanical parameters chosen to describe the pavement material will depend on the type of problem under consideration. For instance, if the time history of the plastic flow of the pavement

material is of interest, then some plastic flow parameters relating permanent strain to the operating stress must be introduced. If the resilient properties or the incipient plastic flow characteristics of pavement materials are of interest, the Young's modulus and the Poisson's ratio of the subgrade and pavement layers are sufficient for a complete description.

FAILURE IN AC PAVEMENTS

The failure of AC pavements generally occurs by two processes: (a) cracking of the bituminous wearing surface and (b) rutting of the wearing surface along the wheel paths.⁹ The fatigue cracking along the wearing surface due to repeated flexural loadings is determined by the magnitude of the tensile strain at the bottom of the wearing surface, while the rutting of the wearing surface may be governed in part by the vertical compressive strain at the top of the subgrade and by the flow of the AC material. Therefore, in this study, the allowable load-carrying capacity and the required overlay thickness for AC pavements will be determined mainly by a limiting vertical compressive strain at the top of the subgrade. However, this may lead to erroneous values because of the gross oversimplifications involved.

For a given load at the pavement surface, the values of the stress and the strain in the pavement and the subgrade depend on the Young's modulus and the Poisson's ratio of the subgrade and each pavement layer. Therefore, if the elastic moduli of the pavement layers are known, it is the Young's modulus of the subgrade that is the unknown parameter determining stress and strain in the pavement and the subgrade. This parameter must be obtained by vibratory nondestructive testing.

FAILURE IN PCC PAVEMENTS

It is assumed herein that PCC pavements fail because of fatigue cracking associated with the repeated flexural stress in the PCC layer. Actually, many failures occur at joints, but this condition is not considered in this study. The fatigue cracking of the wearing surface of PCC pavements is governed by the tensile stress at the bottom of the

wearing surface, and the value of this stress, for a given operating load at the pavement surface, is determined by the elastic moduli of the subgrade and pavement layers.⁹ Assuming that the elastic moduli of the pavement layers are known, it is the subgrade Young's modulus that is the unknown parameter determining the operating value of the tensile stress at the bottom of the PCC layer. This elastic parameter must be supplied by vibratory nondestructive testing.

PAVEMENT EVALUATION AND OVERLAY DESIGN

The computer program PAVEVAL was written to incorporate the material parameters and the limiting stress and strain criteria into a procedure for calculating the allowable load-carrying capacity and the overlay thickness required for pavement upgrading. PAVEVAL, used in conjunction with the computer program SUBE that predicts the value of the subgrade Young's modulus, was developed to be a practical tool for the pavement engineer to use for evaluation and overlay design purposes. Detailed descriptions and listings of the computer programs SUBE and PAVEVAL are given in Appendixes A and B, respectively. Figure 2 gives a flow diagram of the general procedure used for pavement evaluation and overlay design.

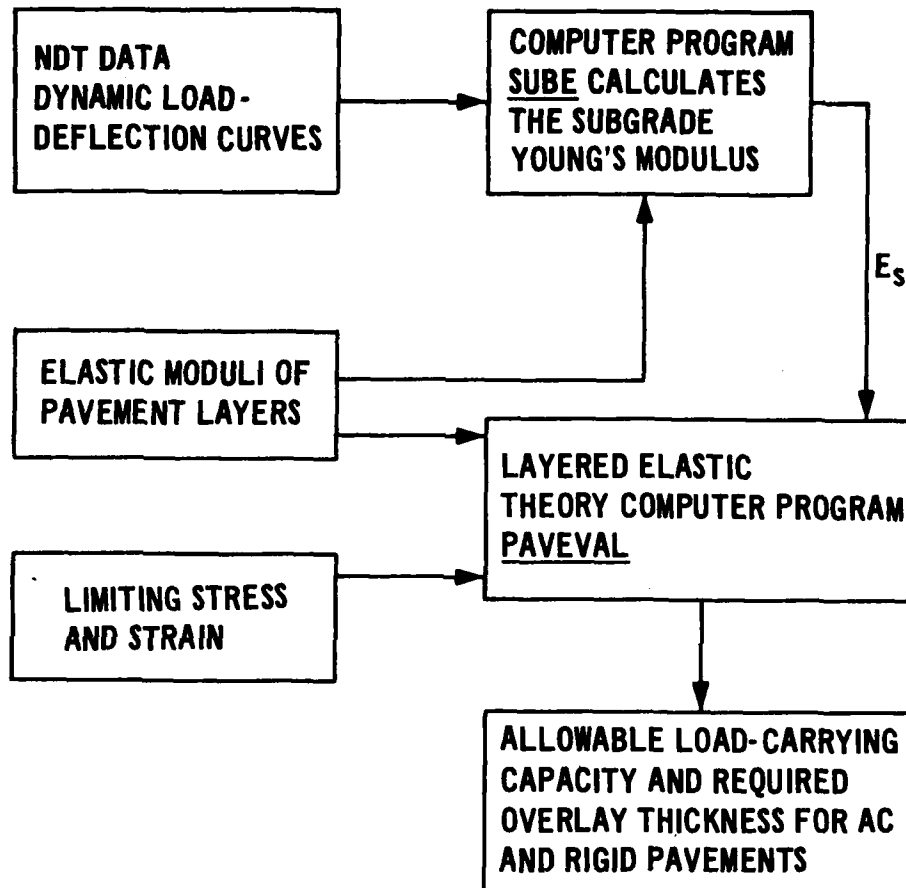


Figure 2. Pavement evaluation and overlay design by the combined methods of layered elastic theory and vibratory nondestructive testing

SELECTION OF THE PAVEMENT AND AIRCRAFT PARAMETERS FOR THE COMPUTER PROGRAMS SUBE AND PAVEVAL

GENERAL CONSIDERATIONS

The pavement parameters required by the computer programs SUBE and PAVEVAL are the Young's modulus, the Poisson's ratio, the thickness of the pavement layers and the subgrade, and the flexural strength for the PCC layers. Some progress has been made toward determining all of the elastic moduli of the pavement layers by vibratory nondestructive testing, but the results are not yet reliable.^{10,11} In this study, only the subgrade Young's modulus is obtained by vibratory nondestructive test methods.

Furthermore, the subgrade is assumed to be infinitely thick. Previous work on the design for PCC pavements incorporates a stiff layer 20 ft below the pavement surface.¹² The present study found it unnecessary to incorporate a stiff layer at some arbitrary depth, so the computer programs SUBE and PAVEVAL assume a homogeneous subgrade.

The computer program PAVEVAL requires aircraft characteristics data as well as pavement parameters to calculate the allowable load-carrying capacity and the required overlay thickness of a pavement. These data include the load on one main gear wheel, the total number of main gear wheels, the tire contact area, and the wheel spacings.

PAVEMENT LAYER THICKNESSES

The pavement layer thicknesses are obtained from construction drawings or from measurements of core samples and thicknesses in core holes in the existing pavement if no construction records are available.

POISSON'S RATIO

The Poisson's ratio of the wearing surface and base and subbase courses was chosen according to the rules $\nu = 0.2$ for PCC pavements, $\nu = 0.3$ for AC pavements and AC base materials, and $\nu = 0.35$ for all other base and subbase materials. The Poisson's ratio for all subgrade

soils is taken to be $\nu = 0.35$. Different choices for these variables can be made according to the type of materials present.

YOUNG'S MODULUS

The Young's modulus of the PCC wearing surface of the PCC pavement is taken to be 4.0×10^6 psi. The temperature-dependent Young's modulus of AC pavements and AC base materials is obtained from Figure 3, corresponding to the pavement surface temperature at the time of the vibratory nondestructive testing. The temperature-dependent Young's modulus value is entered into the computer program SUBE to determine the subgrade Young's modulus. In this study, a value of $E = 450,000$ psi for AC pavements and AC base materials (corresponding to a yearly average temperature of 70°F) is entered into the computer program PAVEVAL that is to be compared with the conventional methods. However, in actual practice, a value of the AC Young's modulus is selected from the curve in Figure 3, representing the appropriate seasonal temperature.

The values of the Young's modulus of granular base and subbase materials can be estimated from the structure and composition of these materials. For instance, the laboratory resilient modulus test gives at least approximate values of the Young's modulus and the Poisson's ratio of these materials.¹³⁻¹⁵ A reasonable estimate of the values of Young's modulus of base and subbase materials can be obtained from Table 1. The subgrade Young's modulus that is entered into the computer program PAVEVAL is the Young's modulus value predicted by the computer program SUBE.

If the base and subbase materials are completely unknown, it is possible to use a trial-and-error procedure to obtain the values for the Young's moduli of these materials and the value of the subgrade Young's modulus by using the nonlinear computer program SUBE and the curves in Figure 4.

The procedure for estimating the values of the Young's moduli of the base and subbase courses is as follows:

- a. Select a trial value of the subgrade Young's modulus.
- b. Use Figure 4 to obtain the Young's moduli of the base and subbase courses.

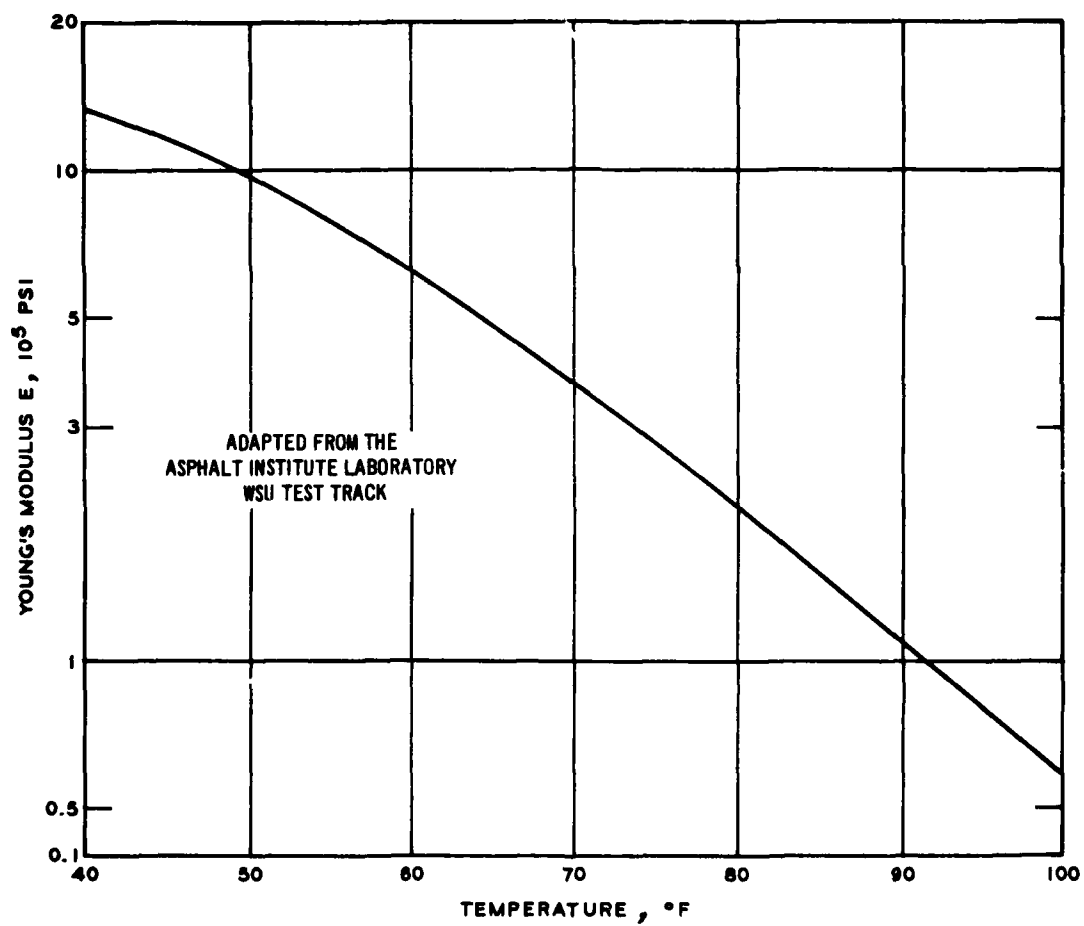


Figure 3. Assumed temperature dependence of Young's modulus of AC pavements and AC base materials

Table 1. Young's Modulus and Poisson's Ratio
of Base and Subbase Materials

Material	Description	Assigned Value	
		of Young's Modulus 10 ³ psi	Assigned Value of Poisson's Ratio
Crushed limestone	Crushed limestone	80	0.35
GW	Well-graded gravel	60	0.35
GW-GM	GW and silty gravel	50	0.35
GP	Poorly graded gravel	40	0.35
GP-GC	GP and clayey gravels	35	0.35
SP	Poorly graded sand	30	0.35
SM	Silty sands, sand silt mixtures	30	0.35
SC	Clayey sands, sand clay	30	0.35
Black base	Mineral aggregate and bituminous material	Temperature dependent	0.30

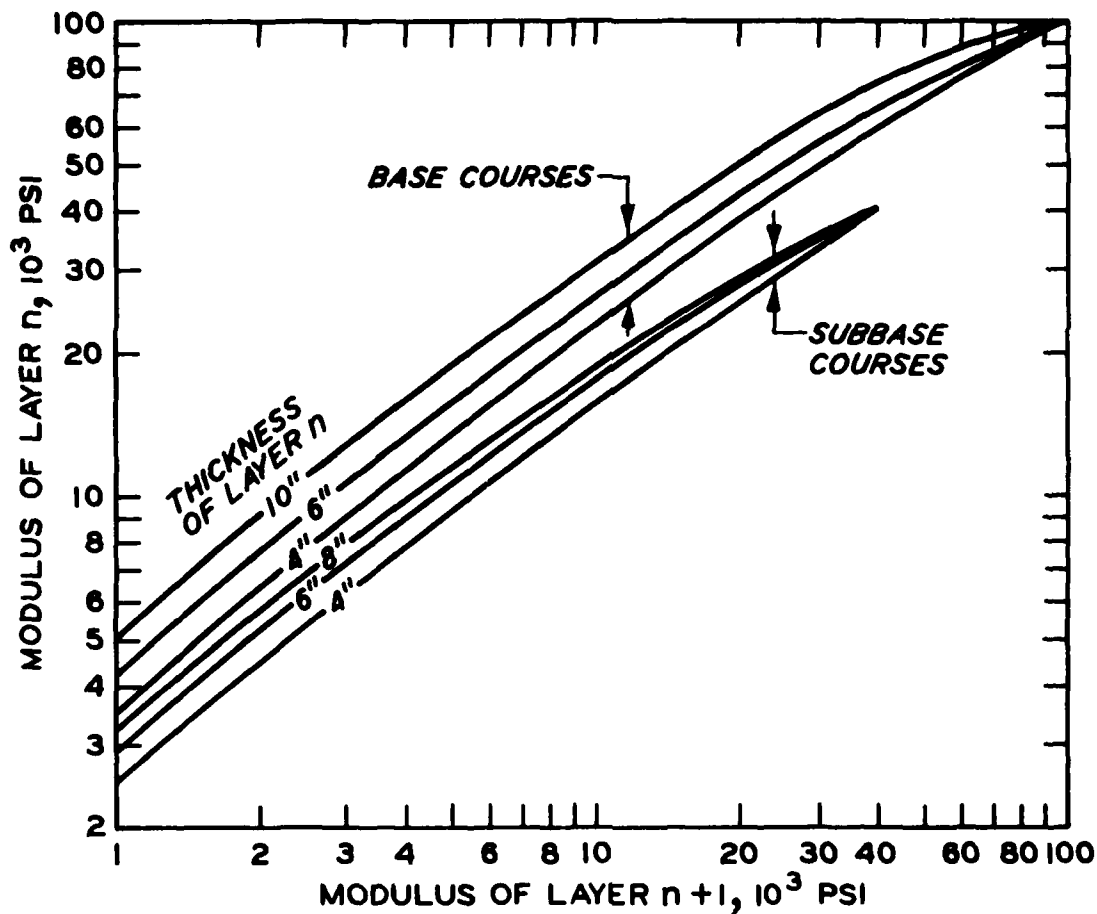


Figure 4. Relationship between the Young's modulus of layer n and the Young's modulus of layer $n + 1$ for various thicknesses of layer n

- c. Place these trial values of the Young's moduli of the base and subbase courses (along with the measured DSM) into the computer program SUBE and get a new value of the subgrade Young's modulus.
- d. Use the new value of subgrade Young's modulus to get new values of the Young's moduli of the base and subbase courses from Figure 4.
- e. Repeat the procedure to the accuracy desired.

FLEXURAL STRENGTH

The flexural strengths of the PCC material of wearing surfaces can be measured in the laboratory on specimens cored from the PCC

pavement.² Splitting tensile tests are conducted, and the results are converted to flexural strengths. The approximate range of variation of the flexural strength of the PCC material found in the wearing surface of PCC pavements is $700 < R < 1020$ psi.¹

AIRCRAFT CHARACTERISTICS

Pavement evaluation and overlay design procedures must include characteristics of the types of aircraft that operate at an airport. Basic aircraft data must be entered into the PAVEVAL computer program in order to calculate the allowable load-carrying capacity and the required overlay thickness for a pavement. The required aircraft data include the load on one wheel, the tire contact area, the total number of main gear wheels, and the transverse and longitudinal wheel spacings. Table 2 gives the required data for several aircraft in common use.

The load on one wheel used in Table 2 takes into consideration the assumption that 5 percent of the gross aircraft weight is supported by the nose wheel. The load on one wheel is therefore given by: gross weight $\times 0.95$ /number of main gear wheels. The operating load on a single wheel is used as the input load in the computer program PAVEVAL to calculate the overlay thickness for PCC and AC pavements. PAVEVAL automatically does the multiple-wheel calculation for the wheel configuration specified by the user.

Table 2. Aircraft Data

Aircraft Gear Configuration or Model Designation	Typical Gross Weight kips	Tire Contact Area sq in.	Total No. of Main Gear Wheels	Load on One Wheel kips	Transverse Wheel Spacing in.	Longitudinal Wheel Spacing in.
Single-wheel	30	190	2	14.25	--	--
Single-wheel	45	237	2	21.38	--	--
Single-wheel	60	271	2	28.50	--	--
Single-wheel	75	297	2	35.63	--	--
Dual-wheel	50	148	4	11.88	--	--
Dual-wheel	75	162	4	17.81	21	--
Dual-wheel	100	170	4	23.75	23	--
Dual-wheel	150	222	4	35.63	26	--
Dual-wheel	200	237	4	47.50	30	--
Dual-tandem	100	99	8	23.75	20	45
Dual-tandem	150	127	8	17.81	20	45
Dual-tandem	200	148	8	23.75	21	46
Dual-tandem	300	198	8	35.63	26	51
Dual-tandem	400	237	8	47.50	30	55
Boeing 727	173	210	4	41.09	34	--
DC-8-63F	358	220	8	42.51	32	55
Boeing 747	778	204	16	46.19	44	58
DC-10-10	433	294	8	51.42	54	64
DC-10-30	558	331	10	53.01	54	64
L-1011	428	282	8	50.83	52	70
Concorde	389	247	8	46.19	26.72	65.7
Boeing 737	111	174	8	13.18	30	--
Lockheed Electra	113	182	4	26.84	26	--
DC-9	115	165	4	27.31	25	--
Convair 880	188	152	8	22.33	22.5	45
Boeing 720	235	188	8	27.91	32	49
Boeing 707	336	218	8	39.90	34	56

LIMITING STRESS AND STRAIN CONDITIONS

As indicated previously, the failure of AC and PCC pavements can be related to limiting strain and stress conditions, respectively. Limiting stress and strain conditions are important for pavement evaluation because they relate the strain in the AC layer and the subgrade of AC pavements, and the stress in the PCC layer of PCC pavements, to the allowable load-carrying capacity and the required overlay thickness of a pavement. Because the stress and strain at any point in a pavement depends on the pavement structure, the limiting stress and strain concept relates the allowable load-carrying capacity and the required overlay thickness to the pavement structure, i.e., to the thickness and the elastic moduli of the pavement layers.

AC PAVEMENTS

The soil of the subgrade will undergo excessive plastic flow under repetitive loads if the repeated vertical strain at the top of the subgrade exceeds a limiting value.⁹ The limiting value of the vertical strain at the top of the subgrade depends on the number of strain repetitions and on the value of the Young's modulus of the soil in the subgrade. Figure 5 gives the limiting vertical strain ϵ_{VL} as a function of the subgrade Young's modulus for 1,200, 6,000, and 25,000 annual strain repetitions.⁹ The curves in Figure 5 are assumed to be valid for all types of subgrade soil and for single- and multiple-wheel loadings.

Figure 6 gives the limiting vertical strain at the top of the subgrade of AC pavements in terms of the total number of load repetitions independent of the value of the subgrade Young's modulus.⁹ A straight-line representation of the data in Figure 6 can be written as

$$\log \epsilon_{VL} = A \log N + B \quad (1)$$

where

ϵ_{VL} = limiting vertical strain at the top of the subgrade
N = total number of load repetitions to failure

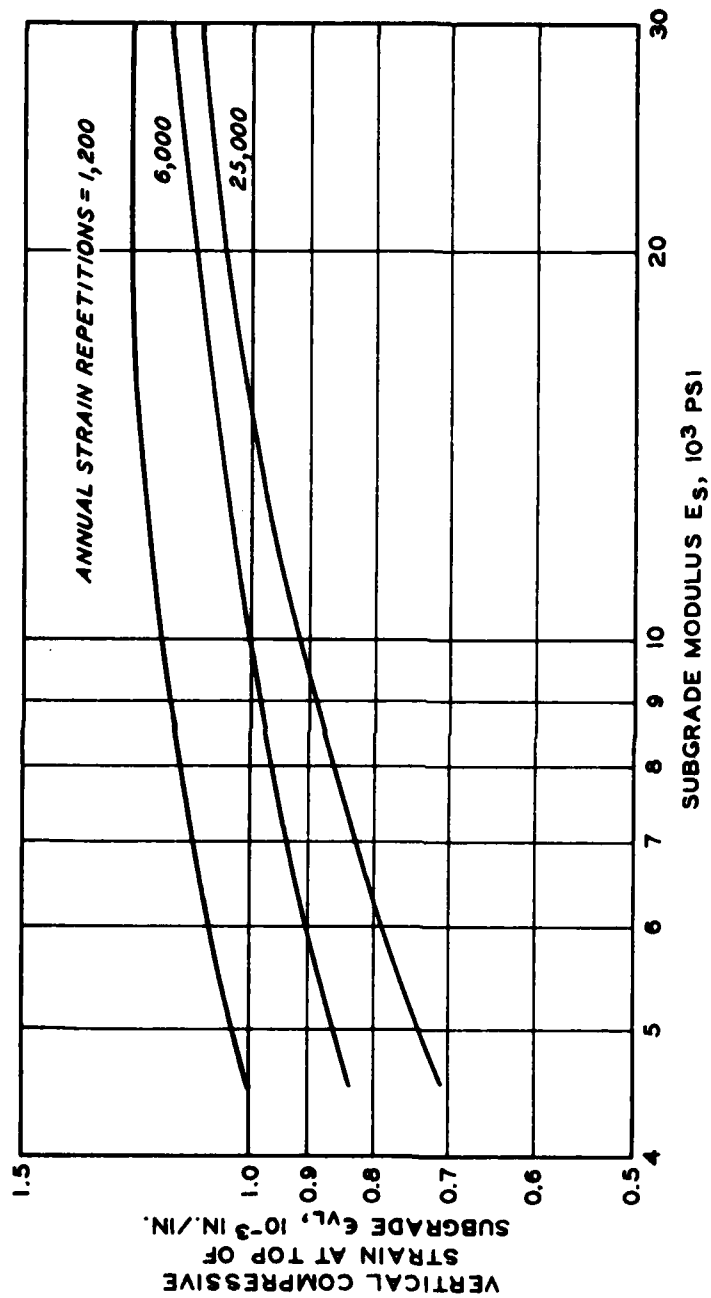


Figure 5. Limiting subgrade strain in terms of the subgrade Young's modulus for conventional AC pavements

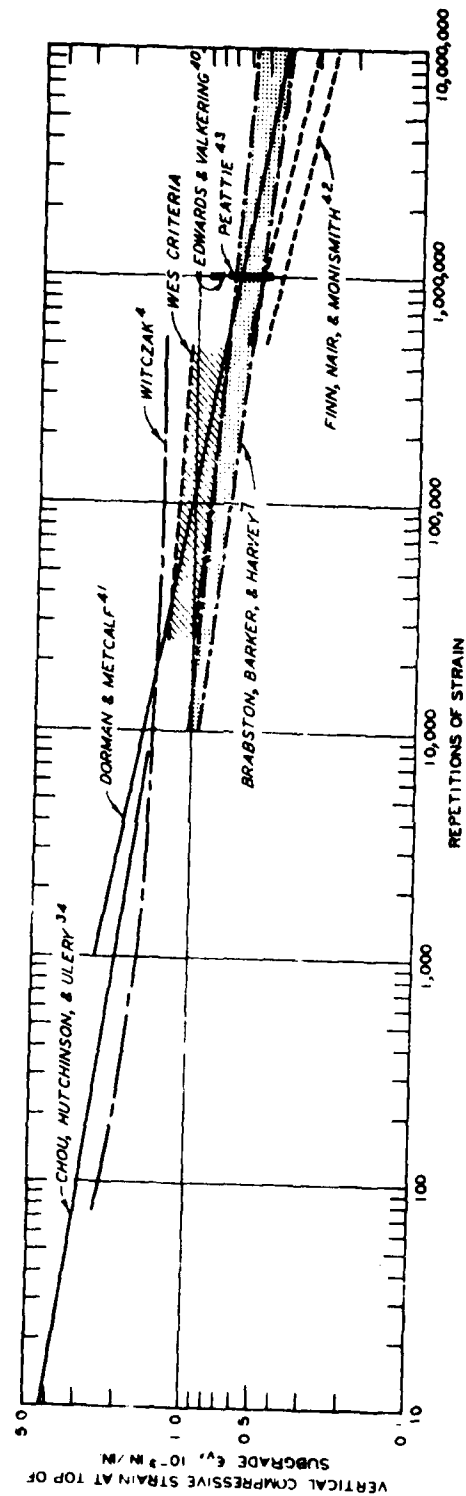


Figure 6. Comparison of subgrade strain criteria (Reference 9)

A best-fit curve through all the data in Figure 6 gives the values, $A = -0.162$ and $B = -2.22$, for the coefficients.

Figure 7 gives the limiting value of the tensile strain ϵ_{RL} at the bottom of the AC layer.⁹ The limiting vertical strain in the subgrade is found to be the controlling condition in most AC pavements, and for all cases considered in this study, it was found that the limiting vertical strain in the subgrade overshadowed the limiting tensile strain in the AC layer.

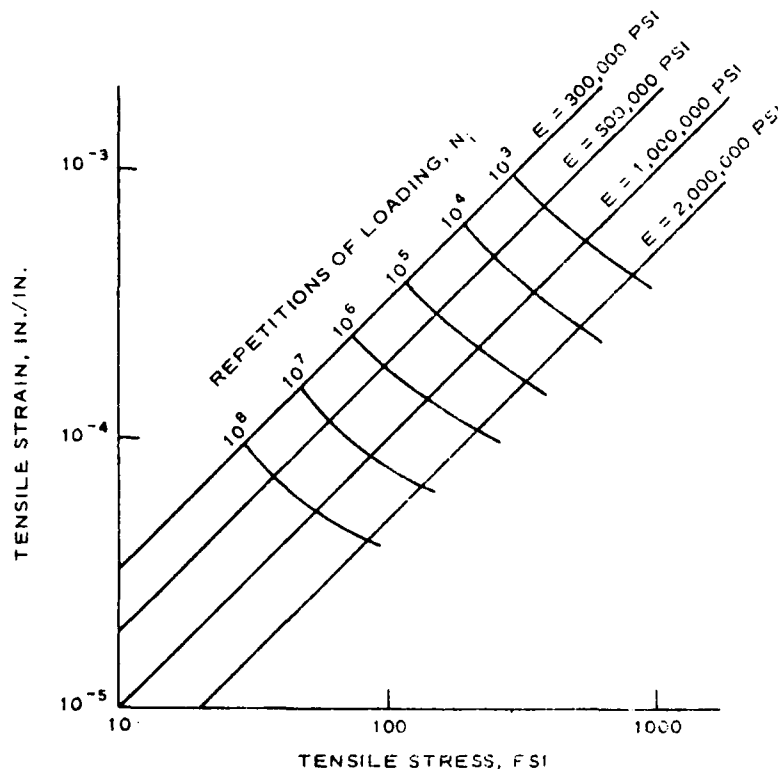


Figure 7. Limiting tensile strain at the bottom of the AC wearing surface

PCC PAVEMENTS

It is assumed that a load applied to the surface of a PCC pavement produces a maximum tensile stress at the bottom of the PCC layer. Further, cracking is assumed to occur first at the bottom of this layer. These are poor assumptions since failure often occurs at

the joints, and the location of the load or of curling conditions is not considered. This incipient cracking will probably be the onset of failure in a PCC pavement; it will begin in the PCC layer when the applied tensile stress at the bottom of this layer exceeds a limiting value of tensile stress.^{8,9} The limiting tensile stress is expressed in terms of the number of load (stress) repetitions and in terms of the flexural strength of the PCC layer as

$$\sigma_{RL} = \frac{R}{A + B \log (COV)} \quad (2)$$

where

σ_{RL} = limiting value of tensile stress, psi

R = flexural strength, psi

A = 0.58901

B = 0.35486

COV = number of coverages

This expression is assumed to be valid whether the stress in the PCC layer is produced by a single- or a multiple-wheel loading.

The number of coverages is related to the number of load repetitions by a factor that depends on the type of aircraft operating on a runway. A coverage refers to a load covering the full width of the traffic lane and thus must be related to the number of repetitions of a particular gear configuration. The connection is made through a pass-to-coverage ratio FAC, which is given by $FAC = N/COV$ where N = number of load repetitions. The limiting radial tensile stress given by Equation 2 can be expressed in terms of the number of load repetitions if the pass-to-coverage ratio is specified. Each gear configuration is associated with a unique value of the pass-to-coverage ratio. Table 3 gives values of the pass-to-coverage ratio for various gear configurations.¹⁶

Table 3. Pass-to-Coverage Ratios

<u>Aircraft Wheel Configuration Type</u>	<u>Ratio (FAC)</u>
Single	5.18
Dual	3.48
Dual-tandem	3.68
L-1011	3.62
B-747	3.70
DC-10-10	3.64
DC-10-30	3.38
DC-8	3.14

SUBGRADE YOUNG'S MODULUS DETERMINED BY VIBRATORY NONDESTRUCTIVE TESTING OF PAVEMENTS

GENERAL CONSIDERATIONS

The basic purpose of the vibratory nondestructive testing of pavements is to supply pavement parameters for the layered elastic theoretical calculation of the allowable load-carrying capacity and the required overlay thickness of a pavement. The layered elastic model of pavements requires the Young's modulus, the layer thickness, and the Poisson's ratio of the subgrade and pavement layers to be known. The elastic moduli of the pavement layers are estimated by various means, and only the subgrade Young's modulus is determined from vibratory non-destructive test results. The subgrade Young's modulus calculated by means of the computer program SUBE serves as an input pavement parameter for the layered elastic theory computer program PAVEVAL that is used for pavement evaluation and overlay design.

Pavements have been noted to behave nonlinearly under dynamic loadings.^{1,3} A nonlinear dynamic layered elastic theory and the computer program SUBE have been developed that determine the subgrade Young's modulus directly from the vibratory nondestructive test data measured at the pavement surface.^{3,4} The input pavement parameters for this dynamic elastic theory are the elastic modulus and the thickness of each pavement layer and the Poisson's ratio of the subgrade. The input from the vibratory nondestructive test data is the dynamic load-deflection curve measured at the surface of a pavement by the WES 16-kip vibrator.

The BISAR computer program is used for the design of PCC and AC pavements^{9,12} and has been modified for pavement evaluation and designated PAVEVAL.

VIBRATORY NONDESTRUCTIVE TEST DATA

The WES 16-kip vibrator applies a static load of 16 kips to the pavement surface and a dynamic load up to 15 kips at frequencies

ranging from 5 to 100 Hz. Both static and dynamic loads are applied to the pavement surface through a circular 18-in.-diam baseplate.

Four types of vibratory nondestructive tests are generally performed on pavements:

- a. Dynamic load-deflection curves that show the dynamic deflection of the pavement surface as a function of the applied load for a fixed frequency of 15 Hz.
- b. Frequency response spectrum that shows the dynamic deflection as a function of frequency for a fixed dynamic load.
- c. Deflection basin measurements.
- d. Rayleigh surface wave dispersion curves that show phase velocity versus wavelength (or frequency).

Only test a above is conducted and used in the method reported to determine the subgrade Young's modulus.

Figure 8 presents a typical dynamic load-deflection curve measured at 15 Hz. The dynamic deflection of the pavement surface is a nonlinear function of the dynamic load applied to the pavement surface. The slope of the dynamic load-deflection curve (tangent modulus) is called the DSM. The numerical value of the DSM is generally obtained from the region of high dynamic loading. Because the dynamic load-deflection curves are nonlinear, a nonlinear dynamic theory is required for their description and to extract the value of the subgrade Young's modulus.^{3,4}

NONLINEAR DYNAMIC THEORY OF PAVEMENT RESPONSE

The nonlinear dynamic theory of pavement response was developed to describe the dynamic load-deflection curves that are measured at the pavement surface and to predict the value of the subgrade Young's modulus from the vibratory nondestructive test measurements.^{3,4} The nonlinear theory of pavement response develops and solves the equation of motion of a nonlinear oscillator and gives a theoretical expression for the dynamic deflection of the pavement surface in terms of the dynamic load applied to the pavement surface. The parameters that describe the nonlinear pavement response are related to the elastic moduli of the pavement layers and the subgrade.³ For a specified

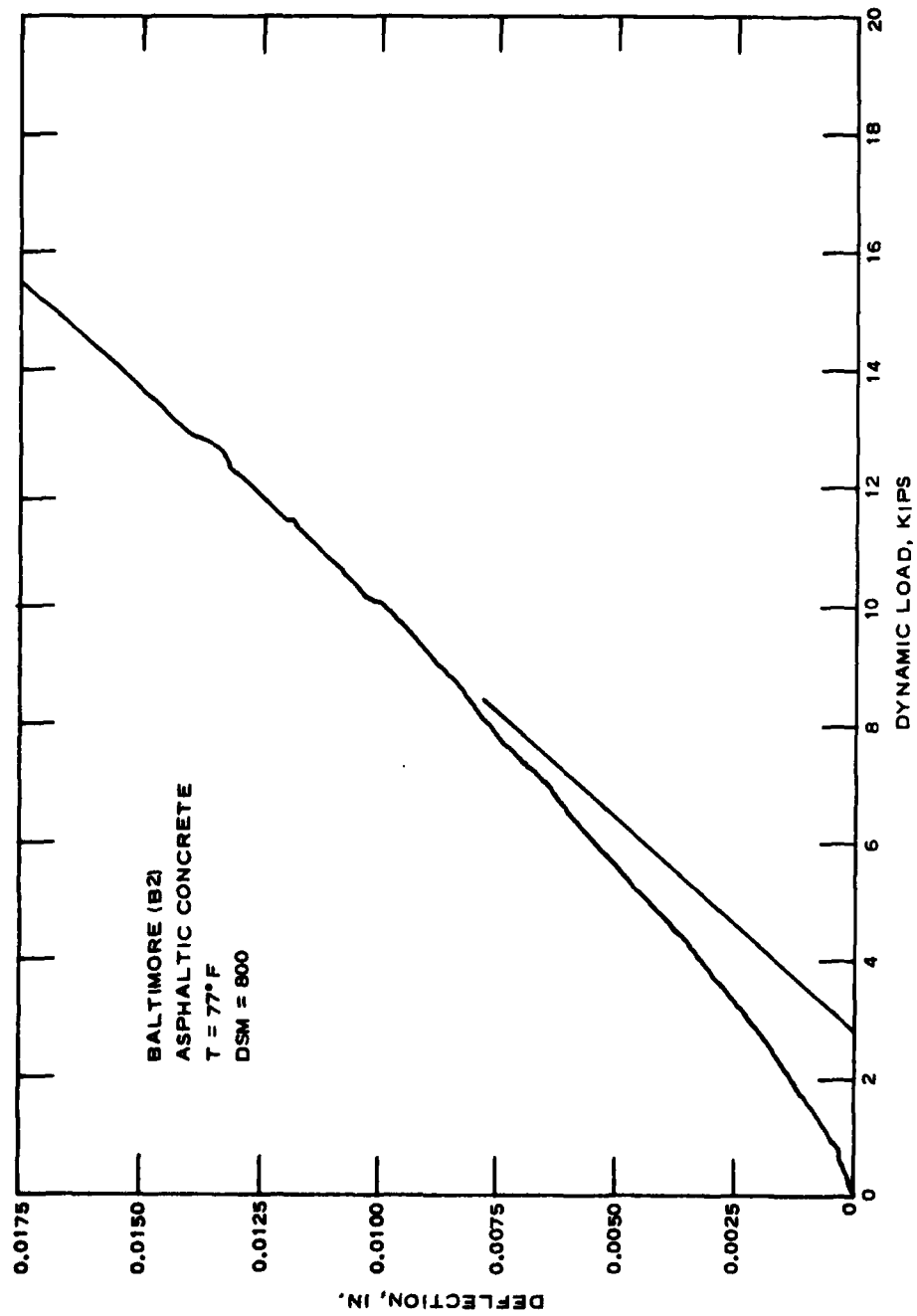


Figure 8. Typical dynamic load-deflection curve for the AC pavement

choice of the elastic moduli of the pavement layers and the Poisson's ratio of the subgrade, the value of the subgrade Young's modulus is obtained by requiring that the theoretically predicted dynamic load-deflection curve agree with the measured dynamic load-deflection curve.^{3,4}

DYNAMIC PAVEMENT RESPONSE COMPUTER PROGRAM SUBE

The computer program SUBE calculates the value of the subgrade Young's modulus from input data taken from the measured dynamic load-deflection curves.⁴ The pavement input parameters for SUBE include the Young's modulus, the Poisson's ratio, and the thickness of each pavement layer, as well as the Poisson's ratio of the subgrade. The computer input that is taken from vibratory nondestructive test data is the DSM value and a point-by-point description of the measured dynamic load-deflection curve. From the DSM value, SUBE calculates the effective mass, the damping constant, the finite depth of influence of the static stress-strain field, and all the other parameters that enter into the nonlinear theoretical model of pavement response.⁴ SUBE iterates the value of the subgrade Young's modulus and determines the value of the subgrade Young's modulus that makes the theoretically predicted DSM value agree with the measured DSM value so that the theoretically predicted dynamic load-deflection curve will agree with the measured dynamic load-deflection curve. Figure 9 outlines the procedure.

NUMERICAL VALUES OF THE PREDICTED SUBGRADE YOUNG'S MODULUS

Tables 4 and 5 show the pavement structures for which dynamic load-deflection curves were measured. These tables also present the values of the elastic moduli of the pavement layers that were used in the computer program SUBE to predict the values of the subgrade Young's modulus. Figure 10 shows a comparison of the subgrade modulus values predicted by the nonlinear dynamic response theory through SUBE and the subgrade modulus values predicted by the formula $E_s = 1500 \text{ CBR}$.¹⁷ The predicted subgrade Young's modulus values depend on the choice of the values of the Young's moduli of the pavement layers. No CBR or

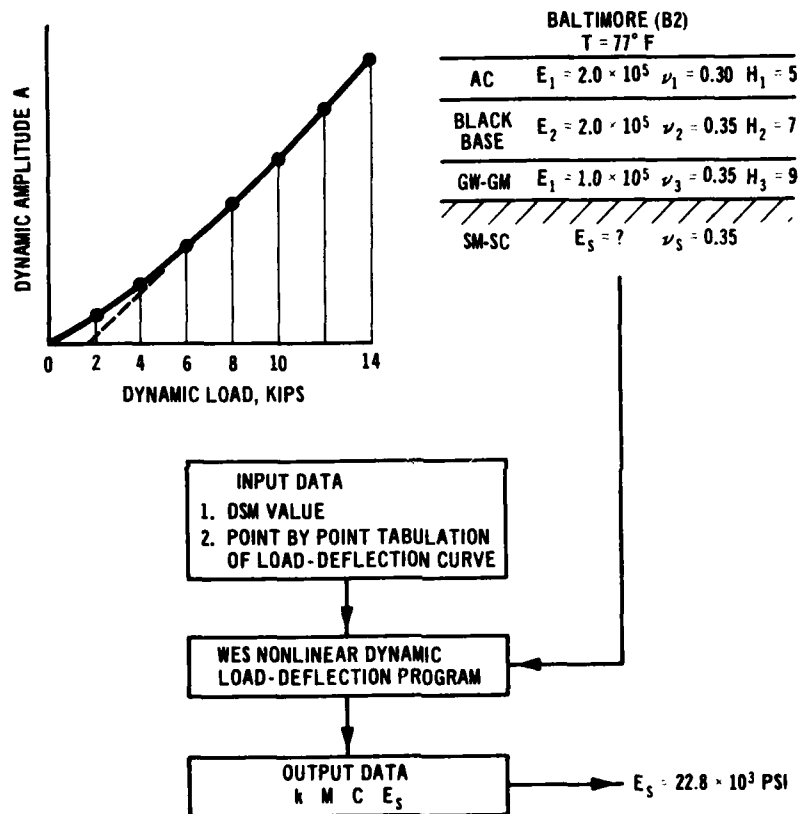


Figure 9. Procedure for obtaining the subgrade Young's modulus from the measured dynamic load-deflection curve

Table 4. AC Pavement Structures Investigated

Site	DSM kips/in.	Wearing Surface				Base				Subbase				Subgrade			
		E ₁ psi	v ₁	h ₁ in.	E ₁ psi	E ₂ psi	v ₂	h ₂ in.	E ₂ psi	E ₃ psi	v ₃	h ₃ in.	E _s WES psi	Nonlinear psi	v _s	CBR	E _s 1500 CBR psi
WES-WEL area subgrade	300	--	--	--	--	--	--	--	--	--	--	--	12,300	0.35	8	12,000	
WES hangar No. 4 subgrade	400	--	--	--	--	--	--	--	--	--	--	--	10,700	0.35	31	46,500	
TETS-adjacent subgrade	320	--	--	--	--	--	--	--	--	--	--	--	27,000	0.35	14	21,000	
TETS-poorhouse subgrade	300	--	--	--	--	--	--	--	--	--	--	--	15,900	0.35	X	X	
TETS-adjacent subgrade	450	--	--	--	--	--	--	--	--	--	--	--	13,000	0.35	8	12,000	
B2A asphaltic concrete	700	230,000	0.3	5	AC T = 77°F	230,000	0.35	7	32,000	0.35	9	9	25,000	0.35	14	21,000	
M18 asphaltic concrete	770	1,400,000	0.3	3.25	AC T = 39°F	34,000	0.35	6.0	GP	--	--	--	29,600	0.35	18	27,000	
WES test area asphaltic concrete	780	100,000	0.3	3.0	AC T = 90°F	30,000	0.35	6.0	Crushed limestone	200,000	0.35	24.0	6,700	0.35	4	6,000	
W1 asphaltic concrete	860	180,000	0.3	9.0	AC T = 87°F	40,000	0.35	5.0	GW-GM	--	--	--	19,500	0.35	20	30,000	
Alum Creek 123-11 asphaltic concrete	820	1,300,000	0.3	7.8	AC T = 41°F	--	--	--	--	--	--	--	19,400	0.35	10	15,000	

(Continued)

Table 4 (Concluded)

Site	DSM kips/in.	Wearing Surface			Base			Subbase			Subgrade		
		E ₁ psi	v ₁	h ₁ in.	E ₂ psi	v ₂	h ₂ in.	E ₃ psi	v ₃	h ₃ in.	E _s WES Nonlinear psi	v _s	E _s 1500 CBR psi
Alum Creek 123-2 asphaltic concrete	880	1,300,000 AC T = 41°F	0.3	7.8	--	--	--	--	--	--	19,000	0.35 GC	10 15,000
W23A asphaltic concrete	980	1,300,000 AC T = 41°F	0.3	3.0	1,300,000 AC base T = 41°F	0.3	3.0	32,000	0.35 GP	7.0	28,000	0.35 SM	18 27,000
Alum Creek 123-23 asphaltic concrete	1000	1,300,000 AC T = 41°F	0.3	7.6	--	--	--	--	--	--	28,000	0.35 GC	17 25,500
Alum Creek 123-32 asphaltic concrete	1230	1,300,000 AC T = 41°F	0.3	7.6	--	--	--	--	--	--	21,000	0.35 GC	17 25,500
B3 asphaltic concrete	1680	700,000 AC T = 58°F	0.3	5.5	700,000 AC base T = 58°F	0.3	5.5	35,000	0.35 GW-GM	10.5	30,000	0.35 SP-SM	17 25,500
W2C asphaltic concrete	1940	300,000 AC T = 73°F	0.3	12.0	60,000 GW-GM	0.35	6.0	--	--	--	35,000	0.35 SP-SC	20 30,000
Fl4A asphaltic concrete	2120	200,000 AC T = 80°F	0.3	14.0	53,000 GP-GC	0.35	5.0	--	--	--	28,700	0.35 GP-GC	20 30,000
Fl3 asphaltic concrete	2780	320,000 AC T = 73°F	0.3	15.0	60,000 GP-GC	0.35	5.0	--	--	--	35,000	0.35 GP-GC	20 30,000
B1 asphaltic concrete	3120	500,000 AC T = 65°F	0.3	5.0	500,000 AC base T = 65°F	0.3	12.5	28,000	0.35 SP-SM	6.0	21,000	0.35 SM	14 21,000
WES test area PCC	3500	4,000,000 PCC	0.2	15.0	200,000 AC base	0.3	6.0	--	--	--	6,800	0.35 Heavy clay	4 6,000

Table 5. Rigid Pavement Structures Investigated

Site	F psi	PCN kips/in.	Wearing Surface			Base			Subgrade				
			Classification	E ₁ psi	v ₁	h ₁ in.	Classification	E ₂ psi	v ₂	h ₂ in.	Classification	E _s psi	v _s
N7	910	1866	PCC	4,000,000	0.2	7.0	E-1(GW)	50,000	0.35	8.0	E-4(SP-SM)	20,000	0.35
N9	724	1920	PCC	4,000,000	0.2	10.0	E-1(GW)	40,000	0.35	8.0	E-4(SP-SM)	11,000	0.35
N11	894	1050	PCC	4,000,000	0.2	7.0	E-7(ML-CL)	16,000	0.35	8.0	E-4(SP-SM)	14,100	0.35
N8	880	1600	PCC	4,000,000	0.2	7.0	E-1(GW)	50,000	0.35	8.0	E4(SP-SM)	21,500	0.35
J5	966	1760	PCC	4,000,000	0.2	8.75	Soil-cement	18,000	0.35	8.0	E7(CL)	14,000	0.35
J3	740	2060	PCC	4,000,000	0.2	7.5	Soil-cement	25,000	0.35	4.75	E7(CL)	28,000	0.35
I20	984	2300	PCC	4,000,000	0.2	10.0	Gravelly sand	26,000	0.35	8.5	E4(SP-SM)	9,000	0.35
N15	564	2640	PCC	4,000,000	0.2	10.0	E-7(ML-CL)	4,500	0.35	9.0	E4(SP-SM)	12,000	0.35
I13	870	3240	PCC	4,000,000	0.2	12.0	Gravelly sand	21,000	0.35	6.0	E6(SP-SC)	17,800	0.35
J1	754	2940	PCC	4,000,000	0.2	9.5	Soil-cement	15,000	0.35	6.25	E-7(CL)	16,000	0.35
I8	1003	3100	PCC	4,000,000	0.2	12.0	Gravelly sand	33,000	0.35	11.5	E-4(SP-SM)	23,300	0.35
H7	929	3480	PCC	4,000,000	0.2	12.0	E-5(OP-SM)	60,000	0.35	5.5	E-6(ML)	23,200	0.35
H1	868	3520	PCC	4,000,000	0.2	14.0	Soil-cement	16,000	0.35	9.0	E-6(ML)	17,200	0.35
H11	894	3700	PCC	4,000,000	0.2	12.5	E-5(SP-SM)	82,000	0.35	11.0	E-6(ML)	21,000	0.35
H3	929	4363	PCC	4,000,000	0.2	12.0	Soil-cement	60,000	0.35	8.0	E-6(ML)	15,000	0.35

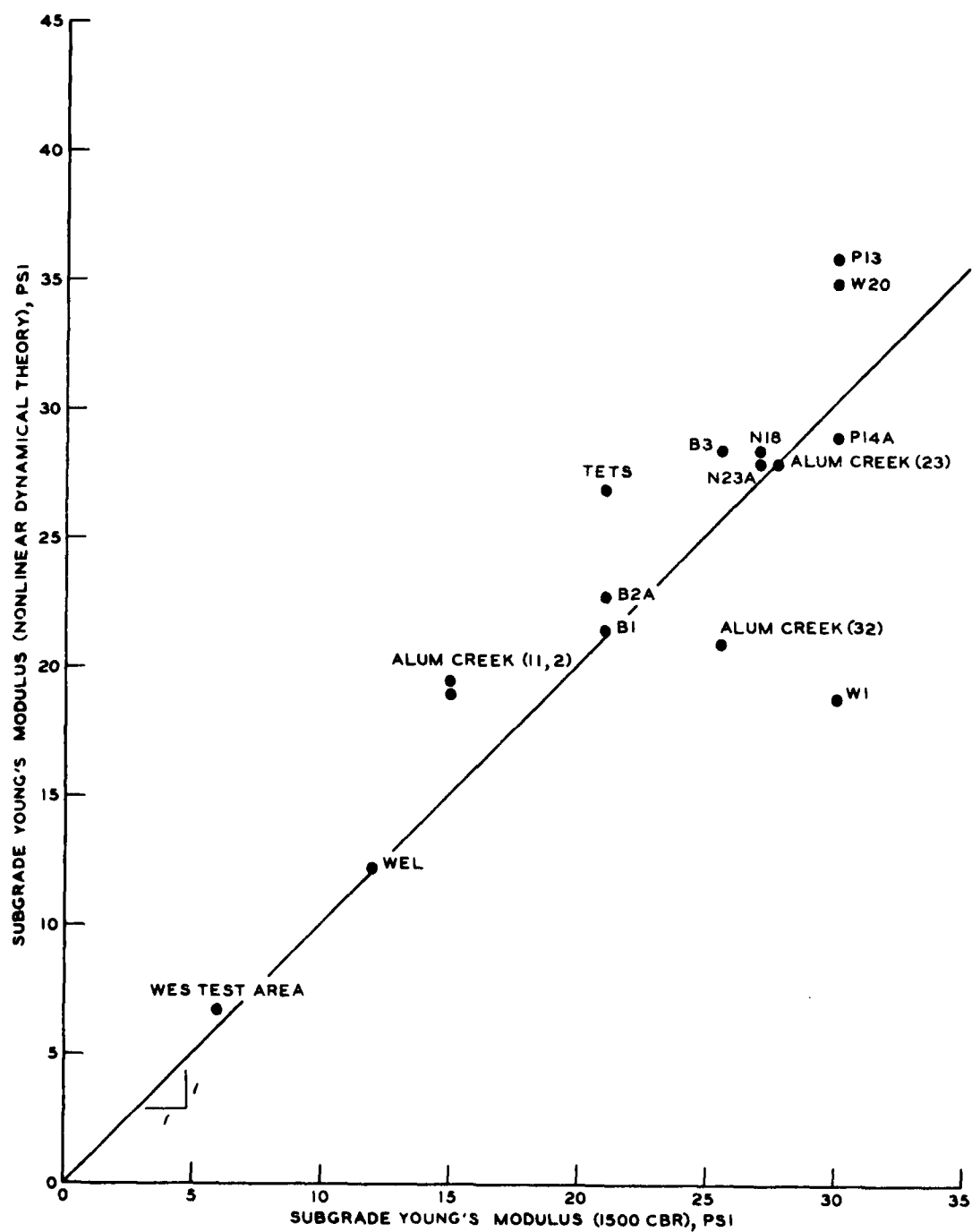


Figure 10. Comparison of subgrade Young's modulus values predicted by the nonlinear dynamic theory computer program SUBE and by the wave propagation formula $E_s = 1500 \text{ CBR}$

coefficient of subgrade reaction values were measured for the subgrade at the PCC pavement sites.

The formula $E_s = 1500 \text{ CBR}$, where E_s represents the subgrade Young's modulus, is obtained as a best-fit straight line through data points for which there was considerable scatter. Therefore, this relationship should be considered to be approximately true, and many deviations from the rule may occur according to the type of materials present and the extreme values of the CBR that may be encountered. The nonlinear dynamic theory of pavement response and the associated computer program SUBE were developed to predict values of the subgrade Young's modulus that are in reasonable agreement with the predictions of the formula $E_s = 1500 \text{ CBR}$. The predicted values of the subgrade Young's modulus can also be compared with laboratory resilient modulus measurements, but this comparison was not made in this study.

Some studies of the sensitivity of the predicted value of the subgrade Young's modulus on the choice of the value of the elastic moduli of the pavement layers have been conducted. Figure 11 shows the dependence of the predicted subgrade modulus values on the values of the Young's moduli of the pavement layers at a pavement site where the DSM value has been measured. The basic pavement structure about which the Young's modulus value of each pavement layer was varied one at a time is as follows:

$$\begin{array}{ll} E_1 = 200,000 \text{ psi} & h_1 = 5.0 \text{ in.} \\ E_2 = 80,000 \text{ psi} & h_2 = 7.0 \text{ in.} \\ E_3 = 40,000 \text{ psi} & h_3 = 9.0 \text{ in.} \end{array}$$

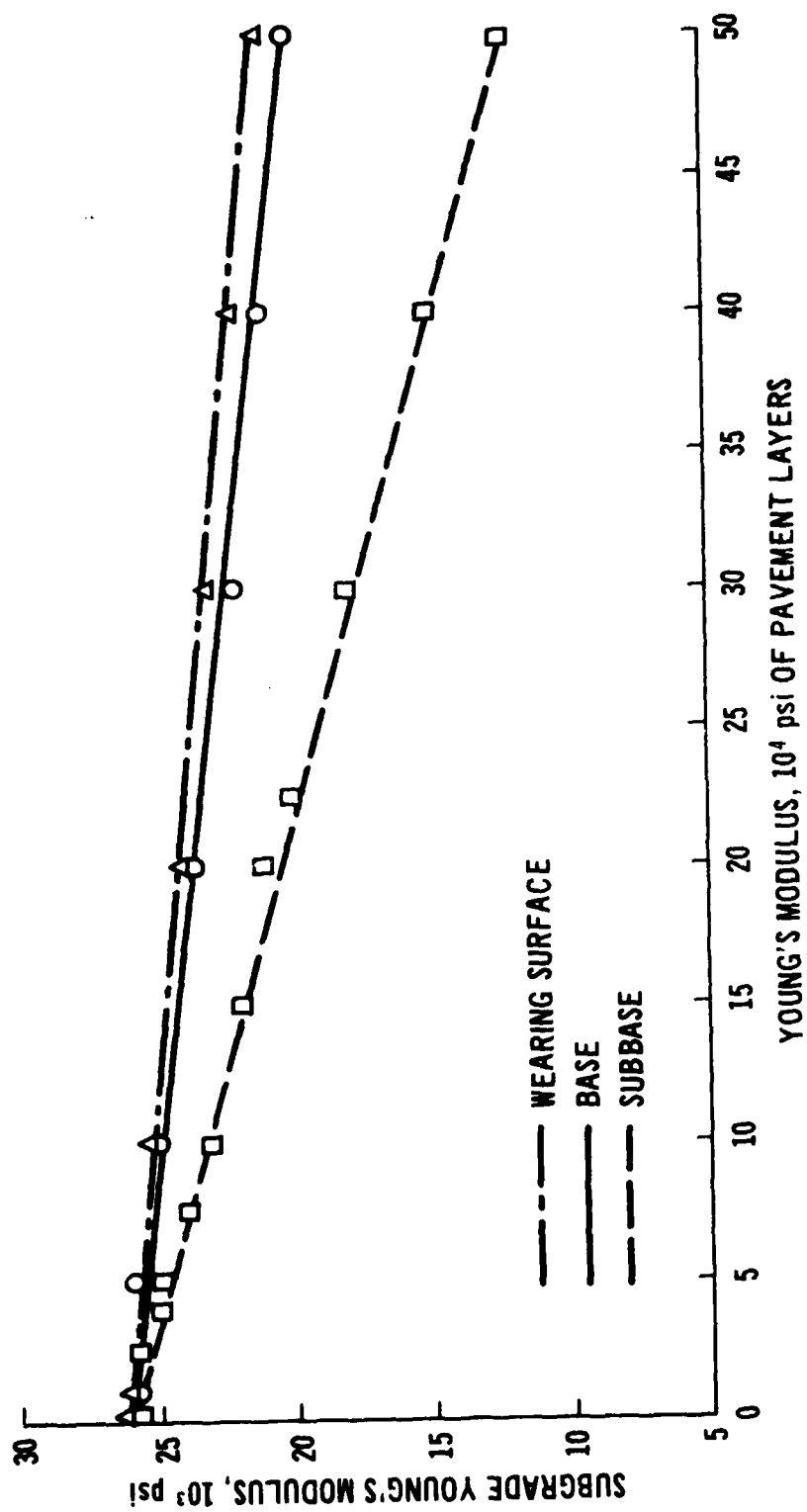


Figure 11. Dependence of the subgrade Young's modulus values predicted by the computer program SUBE on the chosen values of the Young's modulus of the wearing surface and the base and subbase courses

LAYERED ELASTIC THEORY CALCULATION OF THE STRESS AND THE STRAIN IN PAVEMENTS

GENERAL CONSIDERATIONS

The determination of the load-carrying capacity of a pavement and the overlay thickness required to upgrade a pavement entails the calculation of the vertical compressive strain at the top of the subgrade or the tensile strain at the bottom of the AC layer for AC pavements, and the tensile stress at the bottom of the PCC layer for PCC pavements. The calculation of the stress and the strain at points in the pavement and the subgrade is accomplished by modeling the pavement and the subgrade as a semi-infinite layered elastic halfspace for which each layer is described by a Young's modulus, a Poisson's ratio, and a thickness. The layered elastic theory connects the allowable load at the pavement surface and the required overlay thickness for PCC pavements with the limiting values of tensile stress at the bottom of the PCC layer; and for AC pavements, with the compressive vertical strain at the top of the subgrade or the tensile strain at the bottom of the AC layer. The BISAR computer program is used to implement the basic layered elastic theory.

The input parameters for the layered elastic theoretical model of a pavement are the elastic moduli and the thickness of each pavement layer. As discussed previously, the subgrade Young's modulus can be obtained from vibratory nondestructive tests conducted at the surface of the pavement, and the Young's modulus of the wearing surface and the base and subbase courses can be obtained from a classification of the material or from the measured CBR. The thicknesses of the pavement layers are obtained from construction specifications or from measurements in the field. Therefore, all of the parameters required by the layered elastic theory are available for pavement structures.

BISAR COMPUTER PROGRAM

The BISAR computer program was developed by the Shell Oil Company for pavement applications. This computer program calculates the stress

and the strain at any point in the pavement or the subgrade due to a loading at the pavement surface. Particle displacements, stresses, and strains are obtained by numerical integration.

Boundary conditions between the pavement layers may be taken to be rough or smooth. For the rough condition, the radial and tangential stresses and strains are continuous across the layer interfaces. For the smooth condition, the radial and tangential stresses and strains are not continuous across the interface. For PCC pavements, the interface between the PCC wearing surface and the base is assumed to be smooth, while all other interfaces are taken to be rough. For AC pavements, all interfaces are assumed to be rough.

Each pavement layer is characterized by a thickness, a Poisson's ratio, and a Young's modulus. Therefore, three parameters must be specified for each pavement layer. The value of the surface load and the size of the circular loaded area must be specified. The coordinates of the point in the pavement where the stress and the strain are to be calculated must also be specified. The BISAR computer program has the capability of calculating the stress and the strain in the pavement when more than one load is applied to the pavement surface.

The BISAR computer program is modified to calculate the overlay thickness required to upgrade a pavement and the allowable load-carrying capacity of a pavement. The modification consists of an iterative procedure to match the calculated stress and strain with specified limiting values of the stress and the strain. The resulting computer program is called PAVEVAL.

ALLOWABLE LOAD-CARRYING CAPACITY AND REQUIRED OVERLAY THICKNESS OF PAVEMENTS

GENERAL CONSIDERATIONS

The allowable load-carrying capacity and the required overlay thickness of a pavement is related to the pavement structure. The layered elastic theory relates the allowable load-carrying capacity and the required overlay thickness to the pavement structure as represented by the elastic modulus and the thickness of each pavement layer. The following paragraphs describe the layered elastic method of pavement evaluation and overlay design.

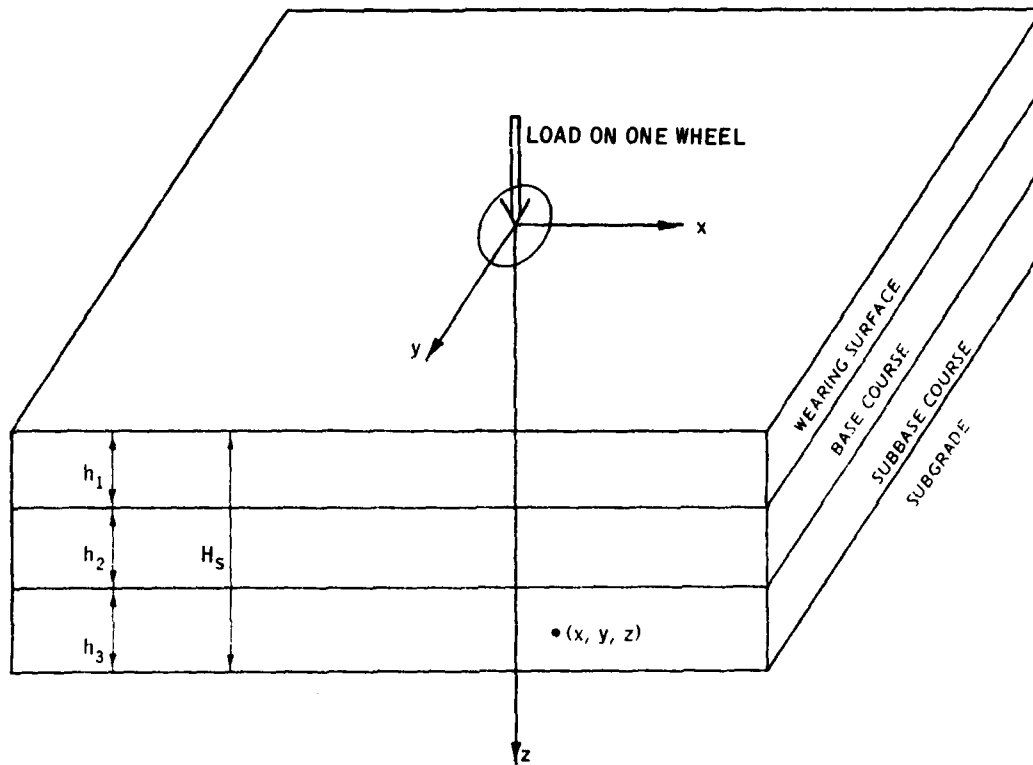
CHOICE OF ELASTIC MODULI FOR PAVEVAL COMPUTER PROGRAM

The value of the subgrade Young's modulus that is used in the PAVEVAL computer program to calculate the allowable load-carrying capacity and the required overlay thickness of a pavement is obtained by using the computer program SUBE to analyze the dynamic load-deflection curves measured at a pavement site. The choice of the elastic moduli of the pavement layers that are entered into PAVEVAL are the same as those selected for SUBE with the exception that the Young's modulus of AC pavements and AC base materials is chosen always to have the value $E = 450,000$ psi in PAVEVAL. This value of the Young's modulus is obtained from Figure 3, corresponding to an assumed average yearly pavement temperature of 70°F.

The values of the Young's modulus of AC pavements and AC base materials that are used in the computer program SUBE to calculate the subgrade Young's modulus are obtained from Figure 3 for a temperature equal to the pavement temperature at the time of the measurement of the dynamic load-deflection curves.

SINGLE-WHEEL LOADING

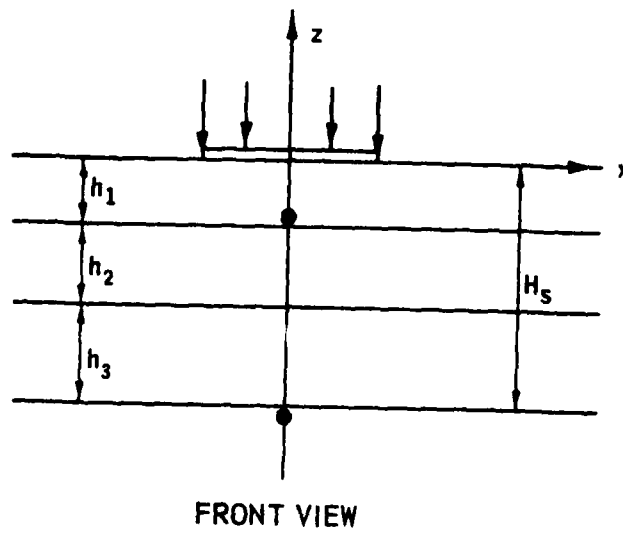
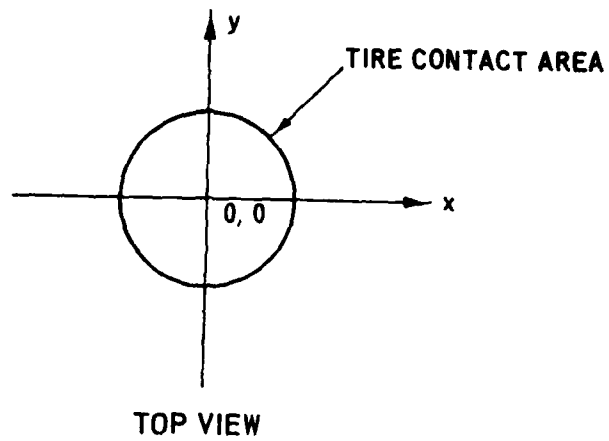
A point in the pavement and the subgrade is designated to have coordinates x, y, z , where x and y describe the horizontal plane and z measures the depth beneath the pavement surface (Figure 12). The



$$H_s = h_1 + h_2 + h_3 = \text{DEPTH TO TOP OF SUBGRADE}$$

Figure 12. Coordinate system chosen for the layered elastic theory of pavements

vertical strain ϵ_V , the tensile strain ϵ_R , and the radial stress σ_R at a point in the pavement are functions of the coordinates of the point in the manner $\epsilon_V = \epsilon_V(x, y, z)$, $\epsilon_R = \epsilon_R(x, y, z)$, $\sigma_R = \sigma_R(x, y, z)$. The maximum values of the stress and the strain in the pavement and the subgrade occur directly beneath the single-wheel load, so that if a coordinate system is chosen whose origin is at the center of the single-wheel load, as shown in Figures 12 and 13, the vertical strain at the top of the subgrade, the radial strain at the bottom of the AC layer, and the tensile stress at the bottom of the FCC layer are represented by



$$\left. \begin{aligned} \epsilon_v &= \epsilon_v(0, 0, H_s) \\ \epsilon_R &= \epsilon_R(0, 0, h_1) \end{aligned} \right\} \text{AC PAVEMENT}$$

$$\sigma_R = \sigma_R(0, 0, h_1) \quad \text{PCC PAVEMENT}$$

Figure 13. Calculation of the stress and the strain for a single-wheel loading

$$\left. \begin{aligned} \epsilon_V &= \epsilon_V (0,0,H_s) \\ \epsilon_R &= \epsilon_R (0,0,h_1) \end{aligned} \right\} \quad (3)$$

$$\sigma_R = \sigma_R (0,0,h_1) \quad (4)$$

where

H_s = depth to the top of the subgrade of an AC pavement

h_1 = thickness of a PCC or an AC wearing surface

The conditions that determine the allowable load-carrying capacity and the required overlay thickness are

$$\left. \begin{aligned} \epsilon_R &= \epsilon_{RL} \\ \epsilon_V &= \epsilon_{VL} \end{aligned} \right\} \quad \text{AC pavement} \quad (5)$$

$$\sigma_R = \sigma_{RL} \quad \text{PCC pavement} \quad (6)$$

where

ϵ_V , ϵ_R and σ_R = values from Equations 3 and 4, respectively

ϵ_{VL} = limiting strain from Figures 5 and 6

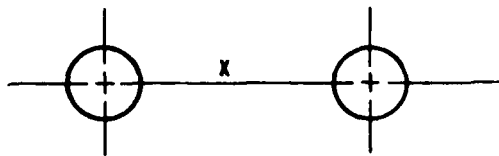
ϵ_{RL} = tensile strain from Figure 7

σ_{RL} = limiting stress from Equation 2

MULTIPLE-WHEEL LOADING

Actual aircraft loadings on a pavement occur through two or more wheels in close proximity. Dual-gear (two wheels) and dual-tandem-gear (four wheels) configurations are commonly used. As indicated in Figure 14, a total number of four main gear wheels are associated with two dual-gear configurations, and eight main gear wheels with two dual-tandem-gear configurations. For the case of multiple wheels, the total strain or stress in the pavement beneath one wheel is due in part to the presence of the other wheels. The maximum values of the stress and the strain at some depth in the pavement occur at a point between the wheels of the gear configuration, but these maximum values of the stress and the strain in the pavement are to a good approximation equal to the values of the stress and the strain in the pavement beneath one of the wheels of a multiple-wheel configuration. The multiple-wheel

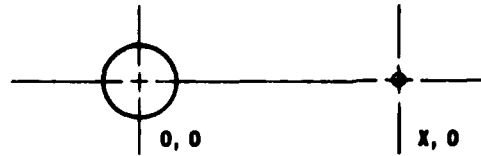
DUAL WHEELS



$$\sigma_{RD} = \sigma_{RD}(0,0,h_1)$$

$$\epsilon_{RD} = \epsilon_{RD}(0,0,h_1)$$

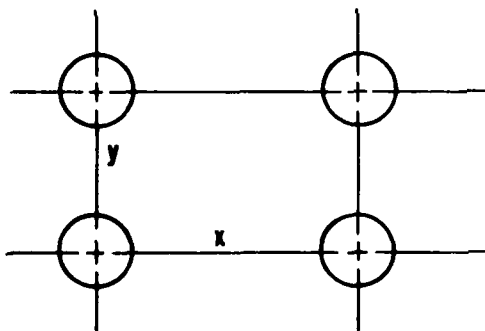
$$\epsilon_{VD} = \epsilon_{VD}(0,0,H_S)$$



PCC PAVEMENT

AC PAVEMENT

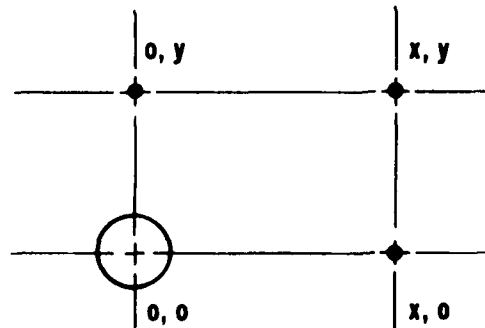
DUAL-TANDEM WHEELS



$$\sigma_{RDT} = \sigma_{RDT}(0,0,h_1)$$

$$\epsilon_{RDT} = \epsilon_{RDT}(0,0,h_1)$$

$$\epsilon_{VDT} = \epsilon_{VDT}(0,0,H_S)$$



PCC PAVEMENT

AC PAVEMENT

Figure 14. Calculation of the total stress and strain for dual and dual-tandem wheels

calculations are made within this approximation. The calculation of the allowable load-carrying capacity and the required overlay thickness must include the additive stress and strain effects associated with multiple-wheel loadings.

The effects of multiple-wheel loadings are accounted for by calculating the net stress and strain in the pavement or the subgrade under a selected wheel and by adding the stress and strain components of the remaining wheels occurring under the selected wheel. The BISAR computer program calculates the stress and the strain in a pavement at any depth directly under one wheel due to the action of the wheel loads applied at the pavement surface. For dual wheels, let

ϵ_{VD} = total vertical strain at a point in the pavement directly under one wheel at the top of the subgrade for AC pavements

ϵ_{RD} = total radial strain under one wheel at the bottom of the AC pavement layer

σ_{RD} = total radial stress at the bottom of the PCC layer at a point under one wheel

For dual-tandem wheels, let

ϵ_{VDT} = total vertical strain at the top of the subgrade for AC pavements at a point directly under one wheel

ϵ_{RDT} = total radial strain at the bottom of the AC pavement layer at a point under one wheel

σ_{RDT} = total radial stress at the bottom of the PCC layer at a point under one wheel

For dual wheels, the conditions that determine the allowable load-carrying capacity and the required overlay thickness are

$$\left. \begin{aligned} \epsilon_{VD}(0,0,H_s) &= \epsilon_{VL} \\ \epsilon_{RD}(0,0,h_1) &= \epsilon_{RL} \end{aligned} \right\} \quad \text{AC pavements} \quad (7)$$

$$\sigma_{RD}(0,0,h_1) = \sigma_{RL} \quad \text{PCC pavements} \quad (8)$$

while for dual-tandem wheels the conditions are

$$\left. \begin{aligned} \epsilon_{\text{VDT}}(0,0,h_s) &= \epsilon_{\text{VL}} \\ \epsilon_{\text{RDT}}(0,0,h_1) &= \epsilon_{\text{RL}} \end{aligned} \right\} \quad \text{AC pavement} \quad (9)$$

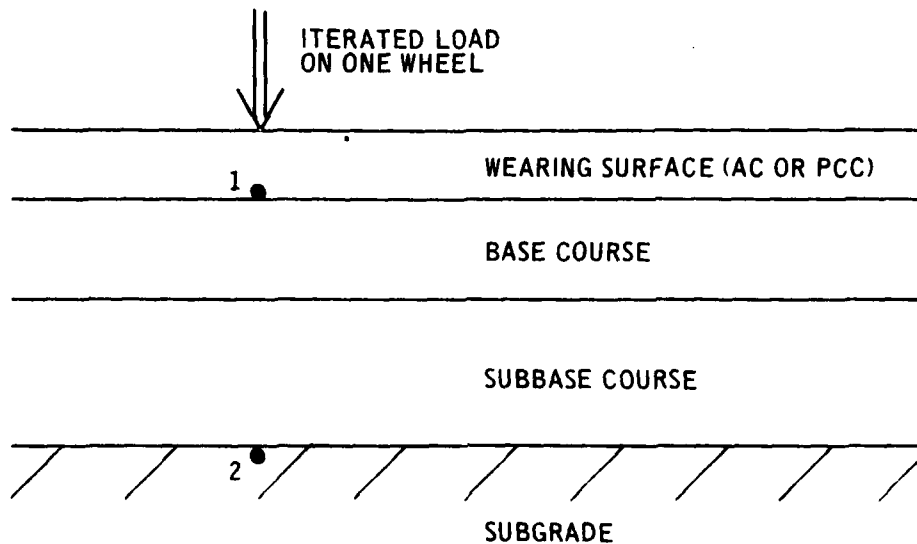
$$\sigma_{\text{RDT}}(0,0,h_1) = \sigma_{\text{RL}} \quad \text{PCC pavement} \quad (10)$$

The limiting stress and strain values do not depend on the type of surface loading and are valid for single- and multiple-wheel loadings.

ALLOWABLE LOAD-CARRYING AND REQUIRED OVERLAY THICKNESS FOR AC AND PCC PAVEMENTS

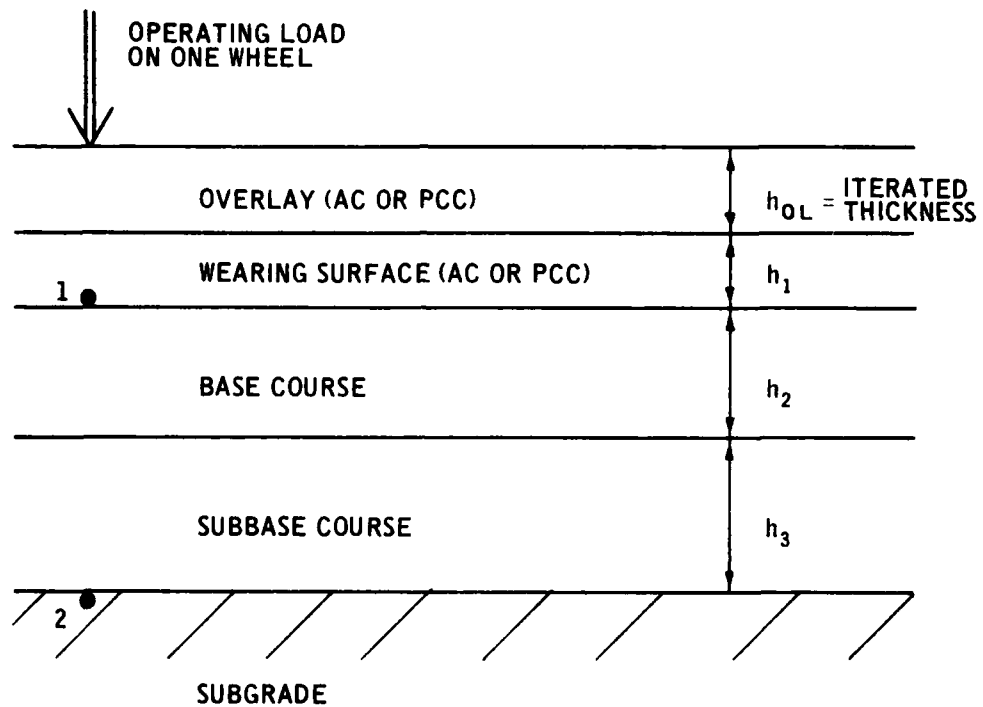
For AC and PCC pavements, the allowable load-carrying capacity is calculated by monitoring the stress and the strain at points in the pavement indicated in Figure 15. The PAVEVAL computer program has the capability of iterating the wheel load at the pavement surface for a given pavement structure until the calculated value of the vertical strain at the top of the subgrade of an AC pavement is equal to the limiting value of the vertical strain or until the calculated value of the tensile strain at the bottom of the AC layer is equal to the limiting value of the tensile strain, as shown in Figures 5, 6, and 7. The calculated value of the tensile stress at the bottom of the PCC layer must equal the limiting value of the tensile stress as given in Equation 2. This determines the allowable load-carrying capacity for AC and PCC pavements.

The required overlay thickness for AC and PCC pavements is calculated by examining the stress and the strain at points in the pavement indicated in Figure 16. The PAVEVAL computer program can be used to iterate the thickness of the overlay for a wheel load until the calculated value of the vertical strain at the top of the subgrade of an AC pavement is equal to or less than the limiting value of the vertical strain and until the calculated value of the tensile strain at the bottom of the AC layer is equal to or less than the limiting value of the tensile strain. For PCC pavements, the calculated value of the tensile stress at the bottom of the PCC layer must equal the limiting value of the tensile stress.



- 1 POINT WHERE TENSILE STRESS AND TENSILE STRAIN IS MONITORED FOR A PCC OR AN AC WEARING SURFACE
- 2 POINT WHERE VERTICAL COMPRESSIVE STRAIN IS MONITORED FOR AN AC WEARING SURFACE

Figure 15. Location of points where the total stress and strain are monitored for the calculation of the allowable load-carrying capacity of AC and PCC pavements



- 1 POINT WHERE TENSILE STRESS AND TENSILE STRAIN IS MONITORED FOR A PCC OR AN AC WEARING SURFACE
- 2 POINT WHERE VERTICAL COMPRESSIVE STRAIN IS MONITORED FOR AN AC WEARING SURFACE

Figure 16. Location of points where the total stress and strain are monitored for the calculation of the required overlay thickness

NUMERICAL VALUES OF THE ALLOWABLE LOAD-CARRYING CAPACITY AND THE REQUIRED OVERLAY THICKNESS

The pavement evaluation procedures discussed previously were applied to a number of PCC and AC pavement structures for single- and multiple-wheel loadings, and the allowable load-carrying capacity and the required overlay thickness were calculated by combining the layered elastic theory with inputs from vibratory nondestructive testing. For these pavement structures, the allowable load-carrying capacity and the required overlay thickness were also calculated by the conventional CBR and DSM methods for AC pavements and by the Westergaard and DSM methods for PCC pavements.

Tables 6-17 present the predicted values of the allowable load-carrying capacity and the required overlay thickness for the AC and PCC pavement sites whose structures appear in Tables 4 and 5. In these tables, the load-carrying capacity is expressed in terms of the load on one wheel. The total allowable airplane load is obtained by the expression: (allowable load on one wheel) \times (total number of main gear wheels)/0.95. Single-wheel, dual-wheel, and dual-tandem-wheel gear configurations were considered. The total number of main gear wheels for these configurations are 2, 4, and 8, respectively. Specific calculations were done for the single-wheel load, the Boeing 727 (dual wheels), the DC-8-63F (dual-tandem wheels), and the DC-10-10 (dual-tandem wheels).

For AC pavements, Figures 17 and 18 compare the layered elastic theory calculation of the allowable load on one wheel and the required AC overlay thickness with the corresponding CBR calculation of these quantities. For PCC pavements, Figures 19 and 20 compare the results of the layered elastic theory calculation of the allowable load on one wheel and the AC and PCC required overlay thickness with the results of the corresponding Westergaard calculations of these quantities. Reference 1 describes the CBR and Westergaard methods of calculating the allowable load and the overlay thickness, respectively, for AC and PCC pavements. The data plotted in Figures 17-21 correspond to the data presented in Tables 6-17.

Table 6. Allowable Load (Layered Elastic Theory Method) of AC Pavement,
1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Tempera- ture* Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	87	72	69	84
W1	860	1080	41	37	36	42
Alum Creek-11	820	541	27	25	15	28
Alum Creek-2	880	581	27	25	24	28
Alum Creek-23	1000	660	32	29	29	33
Alum Creek-32	1230	812	27	25	25	28
W2	1940	2060	79	69	67	79
P14	2120	2630	92	78	75	90
P13	2780	3000	101	85	81	98
N23	980	700	38	36	36	39
B3	1680	1200	88	76	74	88
B1	3120	2680	118	92	81	103
N18	770	630	21	20	20	23

* From Reference 1.

Table 7. Allowable Load (CBR Method) of AC Pavement,
1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Tempera- ture* Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	62	45	34	40
W1	860	1080	45	36	27	38
Alum Creek-11	820	541	10	9.7	5.8	11
Alum Creek-2	880	581	10	9.7	5.8	11
Alum Creek-23	1000	660	20	16	12	19
Alum Creek-32	1230	812	20	16	12	19
W2	1940	2060	76	59	43	59
P14	2120	2630	92	66	50	69
P13	2780	3000	106	74	57	75
N23	980	700	30	23	18	26
B3	1680	1200	65	71	53	72
B1	3120	2680	90	62	45	62
N18	770	630	19	14	10	14

* From Reference 1.

Table 8. Allowable Load (DSM Method) of AC Pavement,
1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature- Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	36	27	22	23
W1	860	1080	50	38	30	39
Alum Creek-11	820	541	28	21	16	21
Alum Creek-2	880	581	27	22	18	23
Alum Creek-23	1000	660	31	24	20	25
Alum Creek-32	1230	812	37	29	24	31
W2	1940	2060	91	79	55	70
P14	2120	2630	120	105	72	92
P13	2780	3000	134	108	78	99
N23	980	700	32	22	19	25
B3	1680	1200	54	66	28	36
B1	3120	2680	122	153	66	82
N18	770	630	30	23	20	25

* From Reference 1.

Table 7. Allowable Load (CBR Method) of AC Pavement,
1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Tempera- ture* Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	62	45	34	40
W1	860	1080	45	36	27	38
Alum Creek-11	820	541	10	9.7	5.8	11
Alum Creek-2	880	581	10	9.7	5.8	11
Alum Creek-23	1000	660	20	16	12	19
Alum Creek-32	1230	812	20	16	12	19
W2	1940	2060	76	59	43	59
P14	2120	2630	92	66	50	69
P13	2780	3000	106	74	57	75
N23	980	700	30	23	18	26
B3	1680	1200	65	71	53	72
B1	3120	2680	90	62	45	62
N18	770	630	19	14	10	14

* From Reference 1.

Table 9. Required Overlay Thickness (Layered Elastic Theory Method) of AC Pavement,
1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Tempera- ture* Adjusted DSM kips/in.	Required Overlay		
			Thickness (Single-Wheel Load) in.	Thickness (Dual-Wheel Load) Boeing 727 in.	Thickness (Dual-Tandem-Wheel Load) in.
			SWL = 35,625 lb	SWL = 41,090 lb	DC-8-63F SWL = 42,510 lb
					DC-10-10 SWL = 51,420 lb
B2	700	805	0.0	0.0	0.0
W1	860	1080	0.0	0.0	0.0
Alum Creek-11	820	541	2.0	4.5	5.0
Alum Creek-2	880	581	2.0	4.5	5.0
Alum Creek-23	1000	660	1.0	3.0	3.5
Alum Creek-32	1230	812	2.0	4.0	5.0
W2	1940	2060	0.0	0.0	0.0
P14	2120	2630	0.0	0.0	0.0
P13	2780	3000	0.0	0.0	0.0
N23	980	700	0.0	0.0	0.0
B3	1680	1200	0.0	0.0	0.0
B1	3120	2680	0.0	0.0	0.0
N18	770	630	5.0	6.5	7.0

* From Reference 1.

Table 10. Required Overlay Thickness (CBR Method) of AC Pavement,
1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Tempera- ture* Adjusted DSM kips/in.	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load)	
					DC-8-63F SWL = 42,510 lb	DC-10-10 SWL = 51,420 lb
B2	700	805	0	0	2.6	1.9
W1	860	1080	0	0.78	3.2	2.9
Alum Creek-11	820	541	4.0	9.8	14	12
Alum Creek-2	880	581	4.0	9.8	14	12
Alum Creek-23	1000	660	3.1	6.1	9.3	8.2
Alum Creek-32	1230	812	3.1	6.1	9.3	8.2
W2	1940	2060	0	0	0	0
P14	2120	2630	0	0	0	0
P13	2780	3000	0	0	0	0
N23	980	700	0.2	3.5	6.4	5.5
B3	1680	1200	0	0	0	0
B1	3120	2680	0	0	0	0
N18	770	630	1.5	6.5	8.4	9.0

* From Reference 1.

Table 11. Required Overlay Thickness (DSM Method) of AC Pavement,
1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Tempera- ture* Adjusted DSM kips/in.	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load)	
					DC-8-63F SWL = 42,510 lb	DC-10-10 SWL = 51,420 lb
B2	700	805	0	4.0	7.3	8.2
W1	860	1080	0	4.1	3.0	2.0
Alum Creek-11	820	541	2.8	5.4	7.9	6.7
Alum Creek-2	880	581	1.7	4.7	7.6	6.2
Alum Creek-23	1000	660	0.8	1.8	6.7	5.6
Alum Creek-32	1230	812	0	1.3	5.0	3.7
W2	1940	2060	0	0	0	0
P14	2120	2630	0	0	0	0
P13	2780	3000	0	0	0	0
N23	980	700	0.4	4.1	7.0	6.2
B3	1680	1200	0	0	5.4	4.8
B1	3120	2680	0	0	0	0
N18	770	630	1.0	4.4	6.9	5.5

* From Reference 1.

Table 12. Allowable Load (Layered Elastic Theory Method) of PCC Pavement,
1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Allowable Load		
			Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips
					DC-8-63F DC-10-10
N7	1866	810	26	22	19 22
N9	1920	724	33	29	26 28
N11	1050	894	24	21	19 20
N8	1600	880	29	24	21 21
J5	1760	966	36	33	28 29
J3	2060	740	26	21	19 21
I20	2300	984	42	37	33 41
N15	2640	864	36	31	28 30
I13	3240	870	54	47	42 45
J1	2940	754	32	29	25 26
I8	3100	1003	66	60	52 56
H7	3480	929	62	56	49 51

(Continued)

Table 12 (Concluded)

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
H1	3520	868	67	57	53	59
H11	3700	894	67	59	53	56
H3	4360	929	59	51	47	50

Table 13. Allowable Load (Westergaard Method) of PCC Pavement,
1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Allowable Load		
			Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips
					DC-8-63F DC-10-10
N7	1866	810	24	15	18 36
N9	1920	724	36	24	24 48
N11	1050	894	24	15	15 33
N8	1600	880	26	18	20 37
J5	1760	966	48	36	38 58
J3	2060	740	26	17	17 34
I20	2300	984	48	38	43 59
N15	2640	864	27	23	21 39
I13	3240	870	55	38	42 59
J1	2940	754	37	25	23 42
I8	3100	1003	66	50	55 69
H7	3480	929	64	48	56 71

(Continued)

Table 13 (Concluded)

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips		Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
						DC-8-63F	DC-10-10
H1	3520	868	81	55		58	75
H11	3700	894	67	50		62	69
H3	4360	929	74	55		62	74

Table 14. Allowable Load (DSM Method) of PCC Pavement,
1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Allowable Load		
			on One Wheel (Single Wheel) kips	on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips
					DC-8-63F DC-10-10
N7	1866	810	35	26	32 40
N9	1920	724	36	26	30 39
N11	1050	894	20	15	18 23
N8	1600	880	30	17	26 34
J5	1760	966	32	24	34 34
J3	2060	740	39	24	34 44
I20	2300	984	43	30	37 48
N15	2640	864	49	33	34 35
I13	3240	870	61	42	44 61
J1	2940	754	55	38	42 58
I8	3100	1003	57	41	44 60
H7	3480	929	65	48	56 71

(Continued)

Table 14 (Concluded)

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel)	Allowable Load on One Wheel (Dual Wheels) Boeing 727	Allowable Load on One Wheel (Dual-Tandem Wheels)	
			kips	kips	DC-8-63F	DC-10-10
H1	3520	868	66	43	43	58
H11	3700	894	66	50	57	75
H3	4360	929	82	58	62	85

Table 15. Required Overlay Thickness (Layered Elastic Theory Method) of ICC Pavement,
1200 Annual Stress Repetition.

Station	Measured BSN Kips/in.	R psi	Overlay Material	Required Overlay Thickness		Required Overlay Thickness		Required Overlay Thickness	
				(Single-Wheel Load) in.	SWL = 35,025 lb	(Dual-Wheel Load) in.	SWL = 41,990 lb	(Single-Wheel Load) in.	(Dual-Wheel Load) in.
									DC-10-10
									SWI = 51,420 lb
N7	1856	810	AC PCC	6.0 4.5		8.5 4.0		13.0 6.5	
N9	1920	724	AC PCC	2.5 1.0		7.5 3.5		9.0 4.0	10.0 4.5
N11	1050	894	AC PCC	6.0 2.5		9.0 4.5		11.0 5.5	13.0 6.0
N8	1600	880	AC PCC	4.0 1.5		7.5 3.5		9.0 4.5	12.0 6.0
J5	1760	966	AC PCC	0.0 0.0		4.5 2.0		7.0 3.0	9.0 4.0
J3	2060	740	AC PCC	5.5 2.5		8.5 4.0		11.0 6.0	13.0 7.0
I20	2300	984	AC PCC	0.0 0.0		X		X	X
N15	2640	864	AC PCC	0.0 0.0		5.5 2.5		7.5 3.0	8.5 3.5

(Continued)

Table 15 (Concluded)

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay		Required Overlay Thickness		Required Overlay Thickness	
				(Single-Wheel Load)	in.	(Dual-Wheel Load)	in.	(Dual-Tandem-Wheel Load)	in.
				SWL = 35,625 lb		SWL = 41,090 lb		DC-8-63F SWL = 42,510 lb	
				DC-10-10 SWL = 51,420 lb					
I13	3240	870	AC	0.0		0.0		0.0	0.0
			PCC	0.0		0.0		0.0	0.0
J1	2940	754	AC	2.5		7.0		9.0	11.0
			PCC	1.0		3.0		4.0	5.0
I8	3100	1003	AC	0.0		0.0		0.0	0.0
			PCC	0.0		0.0		0.0	0.0
H7	3480	929	AC	0.0		0.0		0.0	0.0
			PCC	0.0		0.0		0.0	0.0
H1	3520	868	AC	0.0		0.0		0.0	0.0
			PCC	0.0		0.0		0.0	0.0
H11	3700	894	AC	0.0		0.0		0.0	0.0
			PCC	0.0		0.0		0.0	0.0
H3	4360	929	AC	0.0		0.0		0.0	0.0
			PCC	0.0		0.0		0.0	0.0

Table 16. Required Overlay Thickness (Westergaard Method) of PCC Pavement,
1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay		Required Overlay Thickness	
				Thickness (Single-Wheel Load) in.	Thickness (Dual-Wheel Load) Boeing 727 in.	Thickness (Dual-Tandem-Wheel Load) in.	Thickness (Dual-Tandem-Wheel Load) in.
				SWL = 35,625 lb	SWL = 41,090 lb	DC-8-63F SWL = 42,510 lb	DC-10-10 SWL = 51,420 lb
N7	1866	810	AC PCC	3.5 2.9	13 8.0	11 7.1	7.4 4.9
N9	1920	724	AC PCC	0 0	8.4 6.1	7.8 5.7	1.5 1.7
N11	1050	894	AC PCC	4.0 3.2	12 7.1	13 7.8	8.4 5.6
N8	1600	880	AC PCC	2.5 2.3	10 6.4	8.5 5.6	6.0 4.3
J5	1760	966	AC PCC	3.3 2.8	2.0 2.1	2.0 2.1	0 0
J3	2060	740	AC PCC	3.3 2.8	12 7.8	14 8.7	8.4 5.8
I20	2300	984	AC PCC	0 0	2.2 2.5	0 0	0 0
N15	2640	864	AC PCC	0 0	8.6 6.2	11 7.6	6.1 4.8

(Continued)

Table 16 (Concluded)

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay		Required Overlay Thickness (Dual-Wheel Load)		Required Overlay Thickness (Dual-Tandem-Wheel Load)	
				Thickness (Single-Wheel Load)	in.	Thickness Boeing 727	in.	Thickness in.	in.
				SWL = 35,625 lb		SWL = 41,090 lb		DC-8-63F SWL = 42,510 lb	
				1b		1b		DC-10-10 SWL = 51,420 lb	
I13	3240	870	AC	0		1.2		0	0
			PCC	0		2.0		0	0
J1	2940	754	AC	3.3		8.5		10	3.9
			PCC	2.8		6.2		7.1	3.6
I8	3100	1003	AC	0		0		0	0
			PCC	0		0		0	0
H7	3480	929	AC	0		0		0	0
			PCC	0		0		0	0
H1	3520	868	AC	0		0		0	0
			PCC	0		0		0	0
H11	3700	894	AC	0		0		0	0
			PCC	0		0		0	0
H3	4360	929	AC	0		0		0	0
			PCC	0		0		0	0

Table 17. Required Overlay Thickness (DSM Method) of PCC Pavement,
1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay		Required Overlay Thickness		Required Overlay Thickness	
				Thickness (Single-Wheel Load) in.	SWL = 35,625 lb	Thickness (Dual-Wheel Load) Boeing 727 in.	SWL = 41,090 lb	Thickness (Dual-Tandem-Wheel Load) in.	SWL = 42,510 lb
									DC-8-63F DC-10-10 SWL = 51,420 lb
N7	1866	810	AC PCC	1.2 1.4		9.0 5.9		7.5 5.1	3.8 3.1
N9	1920	724	AC PCC	0 0		12.5 8.2		12.0 8.0	6.8 5.2
N11	1050	894	AC PCC	7.8 5.3		18 10		18 10	12 7.6
N8	1600	880	AC PCC	2.8 2.5		12 7.4		9.5 6.2	6.5 4.6
J5	1760	966	AC PCC	3.2 3.0		16 10		18.5 11	12 8.1
J3	2060	740	AC PCC	0 0		8.8 6.0		7.5 5.3	3.5 3.0
I20	2300	984	AC PCC	0 0		8.0 5.9		7.5 5.6	3.0 2.8
N15	2640	864	AC PCC	0 0		5.8 4.6		5.8 4.6	1.8 2.0

(Continued)

Table 17 (Concluded)

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay		Required Overlay		Required Overlay Thickness	
				Thickness (Single-Wheel Load) in.	Thickness (Dual-Wheel Load) Boeing 727 in.	Thickness (Dual-Tandem-Wheel Load) in.	Thickness (Dual-Tandem-Wheel Load) in.	DC-8-63F	DC-10-10
				SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb		
I13	3240	870	AC PCC	0 0	0 0	0 0	0 0	0 0	0 0
J1	2940	754	AC PCC	0 0	2.2 2.3	0 0	0 0	0 0	0 0
I8	3100	1003	AC PCC	0 0	0 0	0 0	0 0	0 0	0 0
H7	3480	929	AC PCC	0 0	0 0	0 0	0 0	0 0	0 0
H1	3520	868	AC PCC	0 0	0 0	0 0	0 0	0 0	0 0
H11	3700	894	AC PCC	0 0	0 0	0 0	0 0	0 0	0 0
H3	4360	929	AC PCC	0 0	0 0	0 0	0 0	0 0	0 0

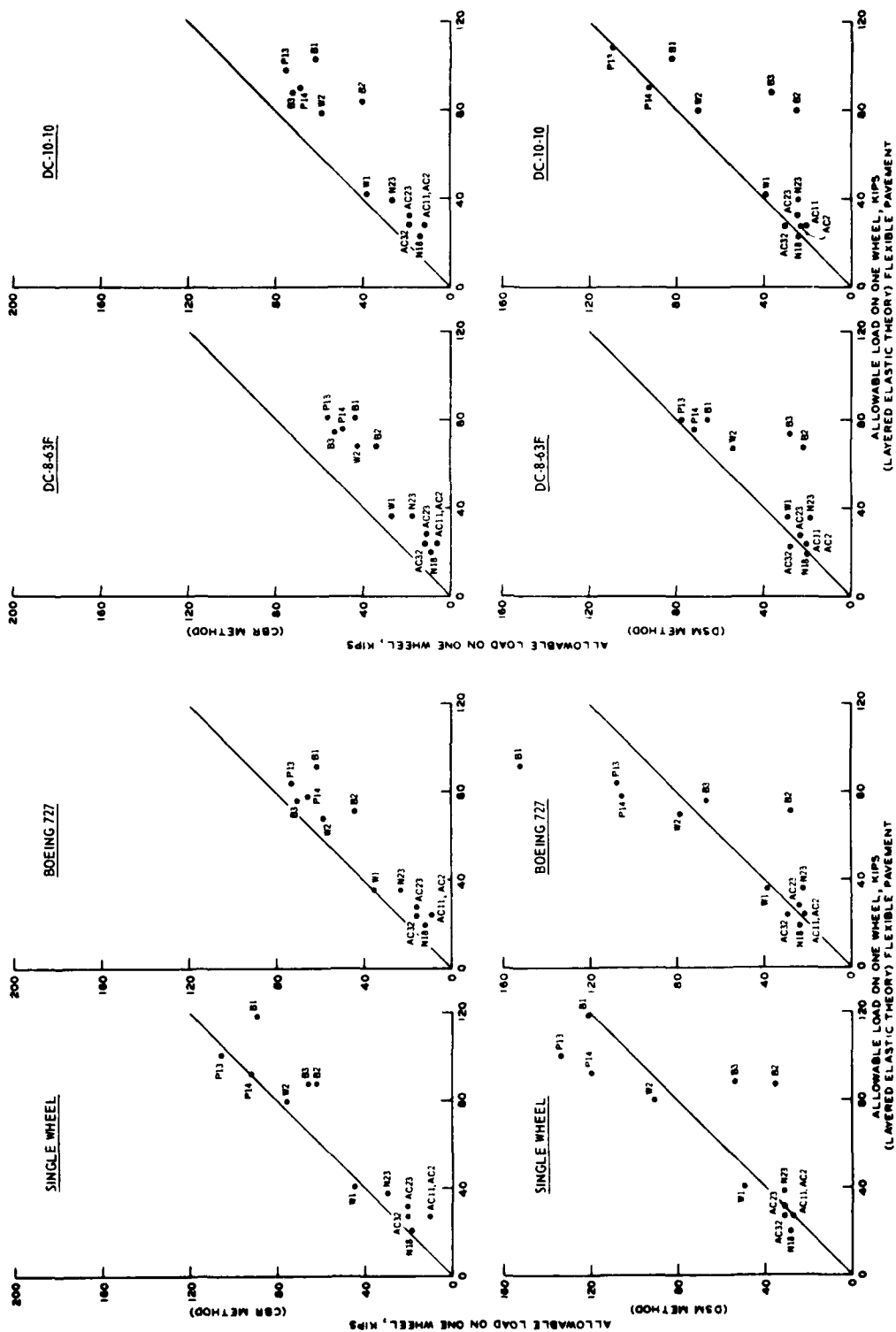
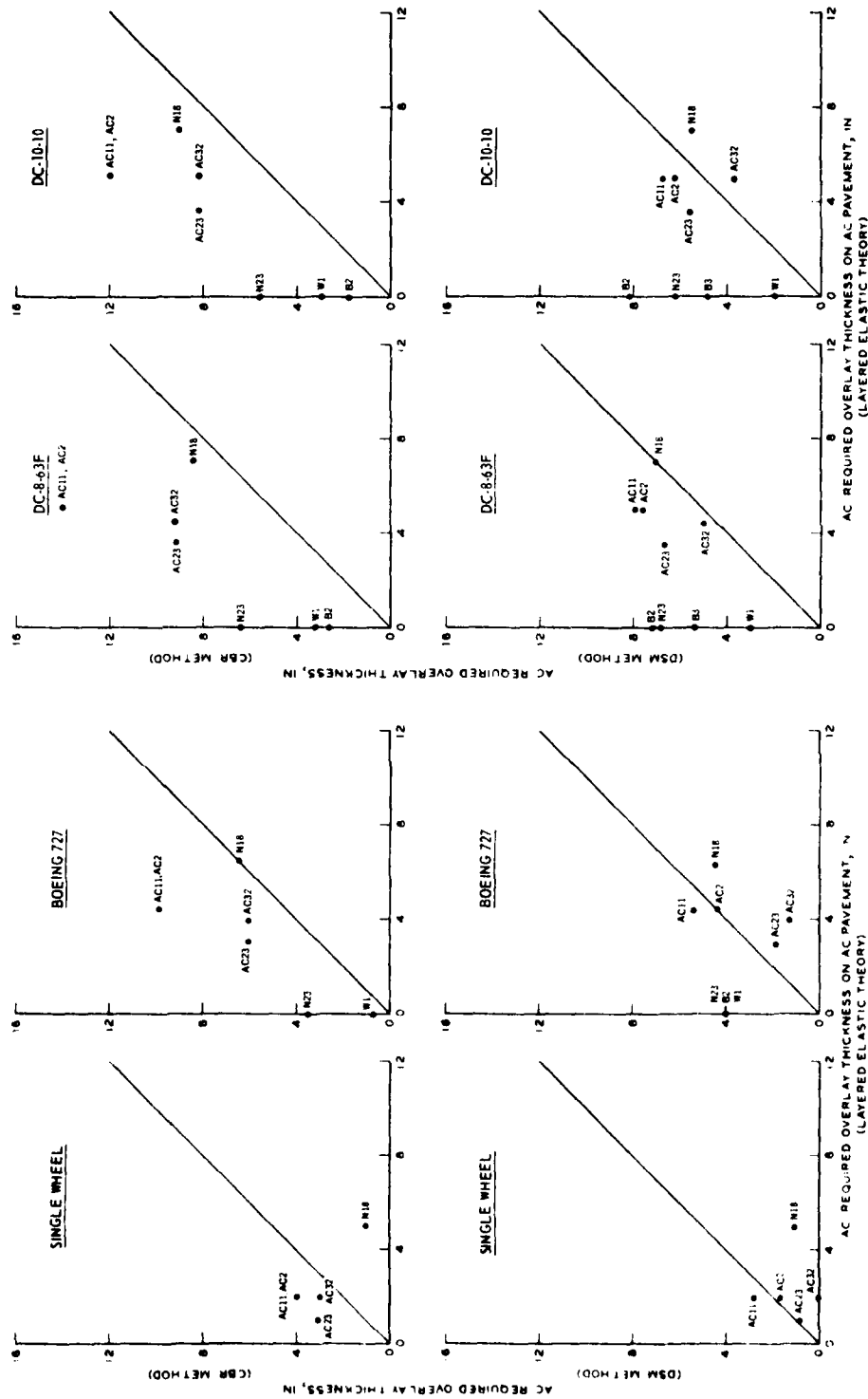
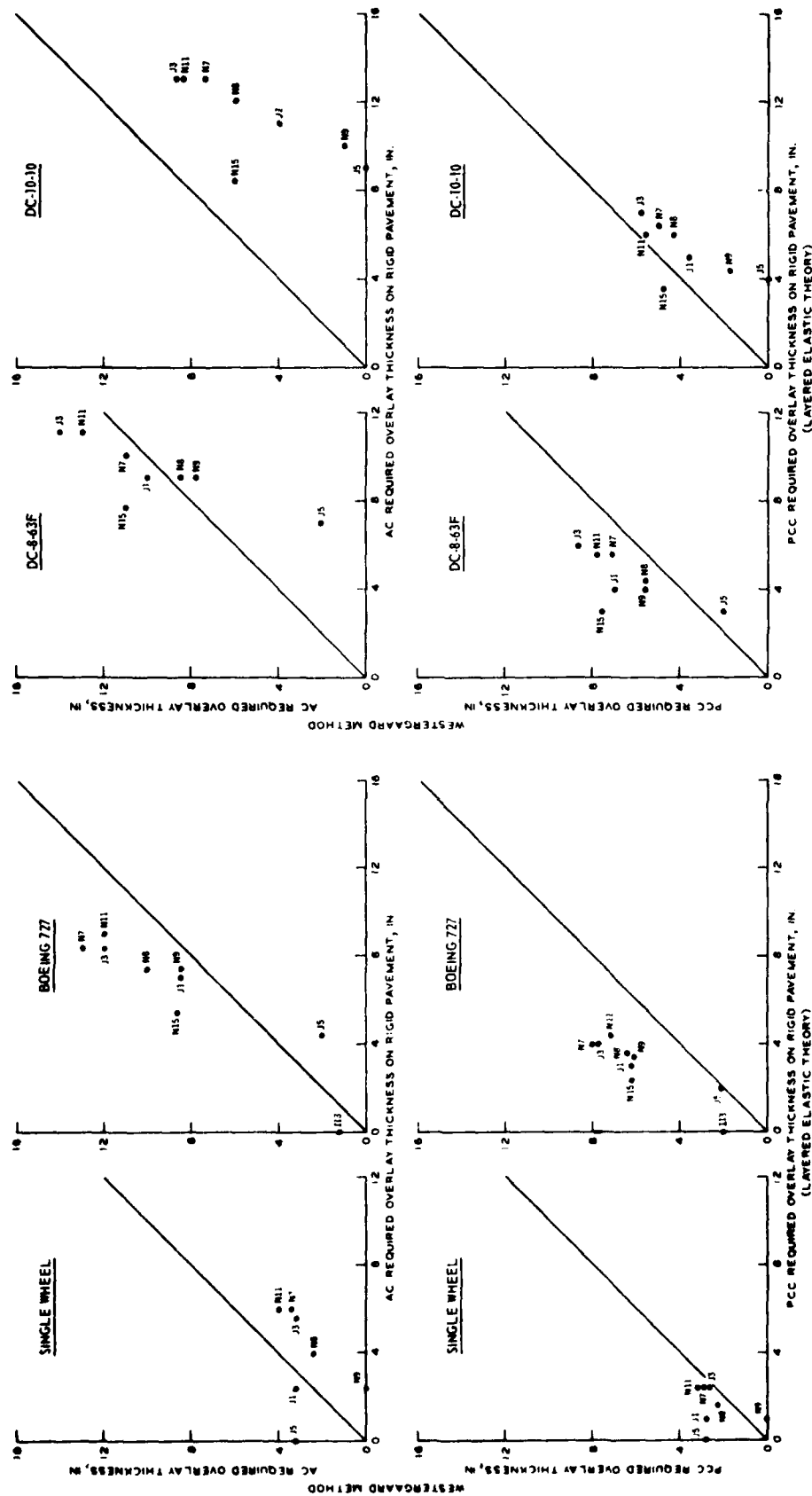


Figure 17. Allowable load-carrying capacity of AC pavements calculated by the layered elastic theory and by the CBR and DSM methods





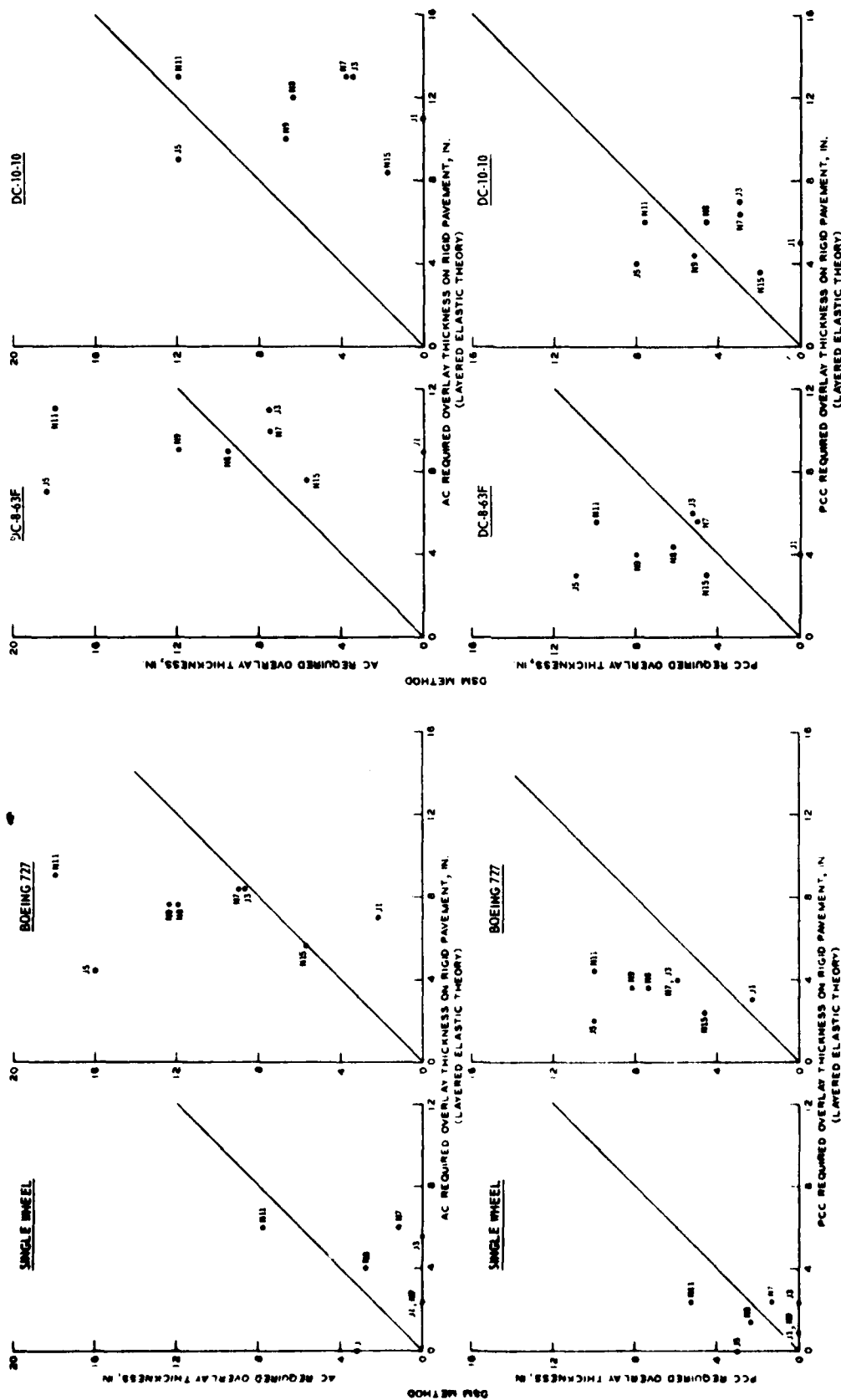
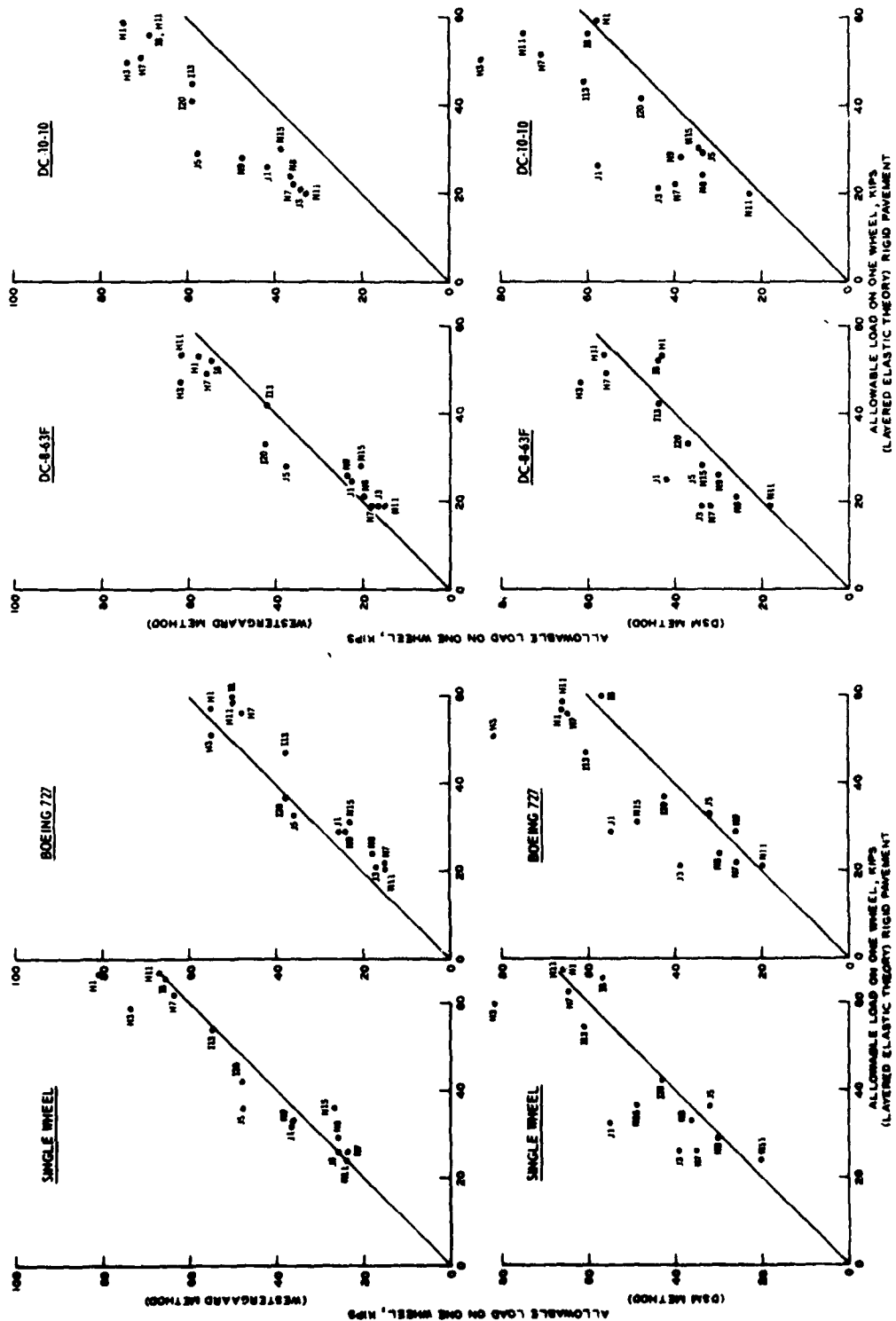


Figure 20. AC and PCC required overlay thicknesses for PCC pavements calculated by the layered elastic theory and the DSM method



For some AC pavements, the layered elastic theory method predicts values of allowable loads that are larger than the aircraft loads, while the DSM method predicts allowable load values that are less than the aircraft load (Tables 6 and 8 and Figure 17). For these cases, the values of the required overlay thicknesses predicted by the layered elastic theory method are zero while those predicted by the DSM method have nonzero values (Figure 18).

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

SUMMARY

The capability of determining the load-carrying capacity of a pavement and the overlay thickness required to upgrade a pavement is important to pavement engineers. A simple method of pavement evaluation combining vibratory nondestructive field tests with a layered elastic theory was developed to satisfy the needs of the pavement engineer. The layered elastic theory approach to calculating the required overlay thickness and the load-carrying capacity of a pavement requires the value of the subgrade Young's modulus, and this value is determined by an analysis of the pavement dynamic response obtained from vibratory nondestructive testing. This approach requires a knowledge of the structure of the pavement and the subgrade as described by the elastic modulus, the Poisson's ratio, and the thickness of each pavement layer.

The method of pavement evaluation presented consists of two parts: (a) the determination of the subgrade Young's modulus from vibratory nondestructive tests that measure the pavement response to an applied dynamic load, and (b) the use of the layered elastic theory and the determined value of the subgrade Young's modulus to predict the allowable load-carrying capacity and the required overlay thickness of a pavement. Two computer programs, SUBE and PAVEVAL, are used to evaluate a pavement based on vibratory nondestructive tests and the layered elastic theory.

The computer program SUBE determines the value of the subgrade Young's modulus from the measured dynamic load-deflection curves and the estimated values of the elastic moduli and thicknesses of the pavement layers. The mathematical model on which SUBE is based is a non-linear harmonic oscillator whose predicted dynamic load-deflection curve is matched to the measured dynamic load-deflection curves in order to determine the value of the subgrade Young's modulus. The predicted values of the subgrade Young's modulus are in essential agreement with the formula $E_s = 1500 \text{ CBR}$ and are not especially sensitive to the choice of the elastic moduli of the pavement layers.

The computer program PAVEVAL calculates the allowable load-carrying capacity and the required overlay thickness values for the layered elastic theory by relating the stress and the strain at any point in the pavement or subgrade to the magnitude of the static load applied to the pavement surface. The elastic moduli, Poisson's ratios, and thicknesses of the pavement layers and the subgrade must be known to use this computer program. For PCC, the flexural strength must also be known. Aircraft parameters including the load on one wheel, the tire contact area, wheel spacings, and the total number of main gear wheels are also required for PAVEVAL.

CONCLUSIONS

The study of predicting pavement performance and overlay design by the combined techniques of layered elastic theory and vibratory nondestructive testing yielded the following conclusions:

- a. The layered elastic theory method using the subgrade Young's modulus determined from the results of vibratory nondestructive tests is sufficient to predict the allowable load-carrying capacity and the required overlay thickness for a pavement; computer programs SUBE and PAVEVAL have been developed to aid in pavement evaluation and overlay design.
- b. The value of the subgrade Young's modulus can be obtained from vibratory nondestructive test results through the use of the computer program SUBE.
- c. Limiting stress and strain criteria can be used in conjunction with the layered elastic theory to determine the allowable load-carrying capacity and the required overlay thickness of a pavement. This can be determined for dual-wheel and dual-tandem-wheel loads, as well as for single-wheel loads, by using the computer program PAVEVAL.

RECOMMENDATIONS

A method has been developed for calculating the allowable load-carrying capacity and the required overlay thickness for pavements by using the combined methods of layered elastic theory and vibratory nondestructive testing. The accuracy of these calculations depends in part on the accuracy of the predicted values of the subgrade Young's

modulus. Further experimental work is necessary to validate the predicted pavement evaluations, overlay designs, and values of the subgrade Young's modulus.

DETERMINATION OF SUBSURFACE STRUCTURE

The determination of the subgrade Young's modulus by the vibratory nondestructive testing technique requires a knowledge of the elastic moduli of the pavement layers above the subgrade. The determination of the allowable load-carrying capacity of a pavement by the layered elastic theory method requires the elastic moduli of all pavement layers as well as the Young's modulus of the subgrade. Therefore, the Young's moduli of the pavement layers are used twice in the procedure for calculating the allowable load-carrying capacity of a pavement. In view of this, it is recommended that:

- a. Vibratory nondestructive tests be developed that will accurately determine the values of the Young's moduli of all pavement layers.
- b. A reliable method be developed to estimate the Young's modulus of the material in each pavement layer in terms of its composition and structure.

STATIC LOAD TESTS

Static load tests are required for the conventional evaluation of PCC and AC pavements. These tests determine the CBR for the AC pavement evaluation and the coefficient of the subgrade reaction for the PCC evaluation using the Westergaard theory. It is recommended that static load tests and vibratory nondestructive tests be performed at a number of pavement sites so that further comparisons can be made.

LABORATORY CONFIRMATION OF FIELD TEST DATA

A complete connection between the resilient modulus laboratory tests and the vibratory nondestructive field tests has not yet been accomplished. However, the results of a preliminary theoretical study show that it is possible to apply a nonlinear dynamic theory to the resilient modulus laboratory test to determine the static elastic

Young's modulus of a subgrade soil and to compare this value with the Young's modulus value predicted by the nonlinear dynamic analysis of the vibratory nondestructive field test data and with the Young's modulus predicted by the formula $E_s = 1500 \text{ CBR}^4$. It is recommended that resilient modulus tests be conducted on undisturbed soil samples taken at sites where vibratory nondestructive tests have been conducted.

REFERENCES

1. Green, J. L. and Hall, J. W., Jr., "Nondestructive Vibratory Testing of Airport Pavements; Evaluation Methodology and Experimental Test Results," Vol 1, Report No. FAA-RD-73-305-I, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1975.
2. Green, J. L., "Literature Review - Elastic Constants for Airport Pavement Materials," Report No. FAA-RD-76-138, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1978.
3. Weiss, R. A., "Nondestructive Vibratory Testing of Airport Pavements; Theoretical Study of the Dynamic Stiffness and Its Application to the Vibratory Nondestructive Method of Testing Pavements," Vol 2, Report No. FAA-RD-73-2-5-II, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1975.
4. _____, "Subgrade Elastic Moduli Determined from Vibratory Testing of Pavements," Report No. FAA-RD-76-158, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1977.
5. Tomita, H., "Field NDE of Airport Pavements; Materials Evaluation," Vol XXXIII, No. 7, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1975.
6. Department of Transportation, Federal Aviation Administration, "Use of Nondestructive Testing Devices in the Evaluation of Airport Pavements," Advisory Circular, AC No. 150/5370-11, Washington, D. C., 1976.
7. _____, "Airport Pavement Design and Evaluation," Advisory Circular, AC No. 150/5320-6B, Washington, D. C., 1974.
8. Hutchinson, R. L., "Basis of Rigid Pavement Design for Military Airfields," Miscellaneous Paper No. 5-7, Department of the Army, Ohio River Division Laboratories, Corps of Engineers, Cincinnati, Ohio, 1966.
9. Barker, W. R. and Brabston, W. N., "Development of a Structural Design Procedure for Flexible Airport Pavements," Technical Report S-75-17, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss., 1975.
10. Lysmer, J., Monismith, C. L., and Watkins, D. J., "Nondestructive Pavement Evaluation by the Wave Propagation Method," Report No. TE-74-2, University of California, Berkeley, Calif., 1974.

11. Vidale, R. F., "The Dispersion of Stress Waves in Layered Media Overlying a Half Space of Lesser Acoustic Rigidity," Ph. D. Thesis, University of Wisconsin, 1964.
12. Parker, F., Barker, W. R., Gunkel, R. C., and Odom, E. C., "Development of a Structural Design Procedure for Rigid Airport Pavements," Report No. FAA-RD-77-81, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1979.
13. Allen, J. A. and Thompson, M. R., "The Effects of Non-Constant Lateral Pressures on the Resilient Response of Granular Materials," Department of Civil Engineering, University of Illinois, Urbana, Ill., 1973.
14. Thompson, M. R. and Robnett, Q. L., "Resilient Properties of Subgrade Soils," Final Report No. UILU-ENG-76-2009, Transportation Research Laboratory, Department of Civil Engineering, University of Illinois, Urbana, Ill., 1976.
15. _____, "Data Summary Resilient Properties of Subgrade Soils," Final Report UILU-ENG-76-2009, Transportation Research Laboratory, Department of Civil Engineering, University of Illinois, Urbana, Ill., 1976.
16. Brown, D. N. and Thompson, O. O., "Lateral Distribution of Aircraft Traffic," Miscellaneous Paper S-73-56, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss., 1973.
17. Heukelom, W. and Foster, C. R., "Dynamic Testing of Pavements," Transactions, American Society of Civil Engineers, Vol 127, Part I, 1962, pp. 425-257.

APPENDIX A: COMPUTER PROGRAM SUBE

The evaluation of rigid and flexible pavements by the combined methods of vibratory nondestructive testing and layered elastic theory requires two computer programs. The computer program SUBE calculates the value of the subgrade Young's modulus from the dynamic load-deflection curves measured at the pavement surface and from the chosen values of the elastic moduli of the pavement layers.

DOCUMENTATION OF THE COMPUTER PROGRAM SUBE

PROGRAM IDENTIFICATION

- a. Program Title. WES Nonlinear Dynamic Load-Deflection Program
- b. Program Code Name. SUBE
- c. Writer. Richard A. Weiss and Adrian P. Park
- d. Organization. U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS 39180
- e. Date. July 1977
- f. Source Language. Fortran IV
- g. Abstract. Program calculates the value of the subgrade Young's modulus by requiring a nonlinear dynamic pavement response model to agree with the measured dynamic load-deflection curves.

ENGINEERING DOCUMENTATION

Narrative Description. The pavement and subgrade are modeled as a nonlinear harmonic oscillator with third-order and fifth-order nonlinear terms. The inertial and damping characteristics of the pavement are introduced by an effective pavement mass and a damping constant. These model parameters are expressed in terms of the measured DSM of the pavement. The elastic characteristics of the pavement and subgrade are represented by the Young's moduli and Poisson's ratios of the pavement layers.

Method of Solution. The solution of a nonlinear harmonic oscillator model of pavement response is determined in terms of the elastic moduli and thicknesses of the pavement layers, the effective

mass of the pavement, the damping constant, and the assorted nonlinear model parameters. The value of the subgrade Young's modulus is obtained by matching the theoretical solution for the dynamic load-deflection curve with the measured value of the dynamic load-deflection curve. The computer program SUBE is used to calculate the subgrade Young's modulus. This computer program iterates the value of the subgrade Young's modulus until the theoretically predicted dynamic load-deflection curve agrees with the measured dynamic load-deflection curve; the value of the subgrade Young's modulus that brings agreement between its measured and theoretical load-deflection curves is the subgrade Young's modulus value that is printed out by the program SUBE.

Program Capabilities. The program calculates the subgrade Young's modulus from dynamic load-deflection curves measured on either flexible or rigid pavements. Rigid and flexible pavements can be handled by entering the appropriate values of the elastic moduli of the wearing surface. The computer program SUBE is valid only for a limited range of measured dynamic load-deflection curves. The DSM is the slope of the measured dynamic load-deflection curve for a dynamic load $F_D = 15$ kips. The computer program SUBE gives valid predictions of the subgrade Young's modulus only within the range $300 < \text{DSM} < 6500$ kips/in.

Printed Output. The printed output consists of the predicted value of the subgrade Young's modulus.

Computer Equipment. The program SUBE was developed on the IBM 360/65 computer.

INPUT GUIDE FOR COMPUTER PROGRAM SUBE

The input for the computer program SUBE is the elastic moduli and thicknesses of the pavement layers, the measured DSM, and a point-by-point description of the measured dynamic load-deflection curve. The point-by-point description of the measured dynamic load-deflection curve is entered into SUBE by means of a data file. The

elastic constants, layer thicknesses, and the measured DSM value are entered into the main body of the computer program. This is done as follows:

2360 Measured DSM value (kips/in.)
2365 Enter 0.0 if dynamic load-deflection curve is straight and
1.0 if dynamic load-deflection curve is curved
5240 Poisson's ratio of layer 1
5250 Poisson's ratio of layer 2
5260 Poisson's ratio of layer 3
5270 Poisson's ratio of layer 4
5300 Young's modulus of layer 1, psi
5310 Young's modulus of layer 2, psi
5320 Young's modulus of layer 3, psi
5330 Initial value of Young's modulus of subgrade, psi
5340 Iteration statement for Young's modulus of subgrade
5580 Thickness of layer 1, in.
5590 Thickness of layer 2, in.
5600 Thickness of layer 3, in.

INPUT GUIDE FOR DATA
FILE FOR SUBE

010	9000	
	Dynamic Load, kips	Dynamic Deflection, mils
012	2	"
014	4	"
016	6	"
018	8	"
020	10	"
022	12	"
024	14	"

SAMPLE PROBLEM USING
PROGRAM SUBE

The calculation of the subgrade Young's modulus using the program SUBE proceeds as follows. From the measured dynamic load-deflection curve, create the following data file and computer input for SUBE.

Date File N23

010 9000
012 2 1.2
014 4 2.5
016 6 3.85
018 8 5.5
020 10 7.5
022 12 9.4
024 14 11.5
SAVE N23

Computer Input

2360 DSMM = 980
2365 CUR = 1.0
5240 POIS1 = 0.3
5250 POIS2 = 0.3
5260 POIS3 = 0.35
5270 POIS4 = 0.35
5300 EMD1 = 1.3×10^6
5310 EMD2 = 1.3×10^6
5320 EMD3 = 4.0×10^4
5330 EMD4 = 10,000
5340 633 EMD4 = EMD4 + 1000
5580 H1 = 3.0
5590 H2 = 3.0
5600 H3 = 7.0
RUN

NAME OF DATA FILE? N23

PROGRAM LISTING

A complete listing of the computer program is presented on the following pages.


```

1* / JOB SUBE,ROKPDH,ROKPDH,OPT=(C,R,T)
2* / LIMIT BAND=100,MIN=60
3* / FTNLX SPACE=90000,FADDMEM=40K,FTNTIME=100000,FTNOPT=(K,M,X)
4* C PROGRAM TITLE WES NONLINEAR DYNAMIC LOAD-DEFLECTION 00000015
5* C CURVE PROGRAM 00000016
6* C PROGRAM CODE NAME SUBE 00000020
7* C WRITER RICHARD A. WEISS AND ARDEN P. PARK 00000025
8* C ORGANIZATION U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION0 00000035
9* C VICKSBURG,MISSISSIPPI 39180 00000045
10* C DATE OCTOBER 1978 00000040
11* C SOURCE LANGUAGE FORTRAN IV 00000045
12* C AVAILABILITY COMPLETE PROGRAM LISTING IS AVAILABLE AT WES0050
13* C 00000055
14* C ABSTRACT COMPUTER PROGRAM SUBE CALCULATES THE VALUE OF THE SUBGRADE00000060
15* C YOUNG'S MODULUS FROM THE DYNAMIC LOAD-DEFLECTION CURVES 00000065
16* C MEASURED AT THE PAVEMENT SURFACE. THIS IS DONE BY 00000070
17* C REQUIRING THE THEORETICAL RESULTS OF A NONLINEAR DYNAMIC 00000075
18* C RESPONSE MODEL FOR THE PAVEMENT TO AGREE WITH THE 00000080
19* C MEASURED DYNAMIC LOAD-DEFLECTION CURVES 00000085
20* C 00000090
21* C PROGRAM SUBE 00000095
22* * POLFT DEC 6, 1972 00000100
23* C 00000105
24* C THE SUBPROGRAM POLFT FITS LEAST-SQUARES POLYNOMIALS TO 00000110
25* C BIVARIATE DATA. IT IS APPLIED TO THE MEASURED DYNAMIC LOAD- 00000115
26* C DEFLECTION CURVES IN THE FORM OF AN ODD ORDER POLYNOMIAL 00000120
27* C HAVING LINEAR,CUBIC AND FIFTH ORDER TERMS 00000125
28* C 00000130
29* C DIMENSION A(15),B(15),S(15),G(15),U(15), 00000135
30* C & ZP(6),RR(5),CR(5), 00000140
31* C & P(100),X(100),Y(100),C(100),Q(100) 00000145
32* C INTEGER FNAME 00000155
33* C 00000170
34* C FNAME=05 00000175
35* C 10 FORMAT(A3) 00000180
36* C 30 GO TO 590 00000185
37* C 36 Z=0 00000190
38* C 0=1 00000195
39* C K=12 00000200
40* C N=N+1 00000205
41* C IF(N.GT.12) GO TO 576 00000210
42* C IF(M.LT.N) GO TO 616 00000215
43* C IF(M.GT.100) GO TO 570 00000220
44* C T7=Z
45* C T8=Z
46* C W7=Z
47* C 305 DO 310 I=1,M 00000230
48* C W7=W7+X(I) 00000235
49* C T7=T7+Y(I) 00000240
50* C 310 T8=T8+Y(I)*Y(I) 00000245
51* C T9=(M*T8-T7*T7)/(M*M-M) 00000250

```

52*	W77=W7/M	00000255
53*	T77=T7/M	00000260
54*	T99=SQR(T9)	00000265
55*	314 FORMAT(///, 'LEAST SQUARES POLYNOMIALS',	
56*	&///, 7X, 'NUMBER OF POINTS =', 12, 7X, 'MEAN VALUE OF X =', 1PE14.6,	
57*	&///, 7X, 'MEAN VALUE OF Y =', 1PE14.6, 7X, 'STD ERROR OF Y =',	
58*	&1PE14.6)	
59*	C	00000295
60*	DO 352 I=1,M	00000300
61*	P(I)=Z	00000305
62*	352 Q(I)=0	00000310
63*	DO 362 I=1,11	00000315
64*	362 A(I)=Z	
65*	B(I)=Z	
66*	S(I)=Z	
67*	E1=Z	
68*	F1=Z	
69*	W1=M	00000330
70*	N4=K	00000335
71*	I=1	00000340
72*	K1=2	00000345
73*	IF(N.NE.0) K1=N4	00000350
74*	380 W=Z	00000355
75*	DO 386 L=1,M	00000360
76*	386 W=W+Y(L)*Q(L)	00000365
77*	S(I)=W/W1	00000370
78*	IF(I-N4.GE.0.OR.I-M.GE.0) GO TO 428	00000375
79*	E1=Z	00000380
80*	DO 398 L=1,M	00000385
81*	398 E1=E1+X(L)*Q(L)*Q(L)	00000390
82*	E1=E1/W1	00000395
83*	A(I+1)=E1	00000400
84*	W=Z	00000405
85*	DO 416 L=1,M	00000410
86*	V=(X(L)-E1)*Q(L)-F1*P(L)	00000415
87*	P(L)=Q(L)	00000420
88*	Q(L)=V	00000425
89*	416 W=W+V*V	00000430
90*	F1=W/W1	00000435
91*	B(I+2)=F1	00000440
92*	W1=W	00000445
93*	I=I+1	00000450
94*	GO TO 380	00000455
95*	428 DO 432 L=1,12,1	00000460
96*	432 G(L)=Z	00000465
97*	G(1)=0	00000470
98*	DO 464 J=1,N	00000475
99*	S1=Z	00000480
100*	DO 448 L=1,N	00000485
101*	IF(L.NE.1)G(L)=G(L)-A(L)*G(L-1)	00000490
102*	IF(L.GT.2)G(L)=G(L)-B(L)*G(L-2)	00000495
103*	448 S1=S1+S(L)*G(L)	00000500
104*	U(J)=S1	00000505

105*	L=N	00000510
106*	DO 460 I2=2,N	00000515
107*	G(L)=G(L-1)	00000520
108*	L=L-1	00000525
109*	460 CONTINUE	00000530
110*	G(1)=Z	00000535
111*	464 CONTINUE	00000540
112*	T=Z	00000545
113*	DO 488 L=1,M	00000550
114*	C(L)=Z	00000555
115*	J=N	00000560
116*	DO 482 I2=1,N	00000565
117*	C(L)=C(L)*X(L)+U(J)	00000570
118*	J=J-1	00000575
119*	482 CONTINUE	00000580
120*	T3=Y(L)-C(L)	00000585
121*	T=T3*T3	00000590
122*	488 CONTINUE	00000595
123*	IF(M.NE.N) GO TO 496	00000600
124*	T5=0	00000605
125*	GO TO 498	00000610
126*	496 T5=T/(M-N)	00000615
127*	498 Q7=1-T/(T9*(M-1))	00000620
128*	IF(M.NE.99)GOTO789	00000625
129*	ITEMP=N-1	
130*	PRINT 500,ITEMP,Q7	
131*	500 FORMAT(/,' POLYFIT OF DEGREE ',I2,' INDEX OF DETERM = ',	
132*	81PE14.6)	
133*	PRINT 516	00000645
134*	516 FORMAT(/,' TERM',8X,'COEFFICIENT',/)	
135*	DO 526 J=1,N	00000655
136*	I2=J-1	00000660
137*	526 PRINT 527,I2,U(J)	00000665
138*	527 FORMAT(14,7X,1PE14.7)	00000670
139*	PRINT 530	00000675
140*	530 FORMAT(/,' X-ACTUAL',12X,'Y-ACTUAL',3X,'Y-CALC',8X,'DIFF',	
141*	89X,'PCT-DIFF',/)	
142*	DO 550 L=1,M	00000690
143*	Q8=Y(L)-C(L)	00000695
144*	IF(C(L)-0.0) 540,548,540	00000700
145*	540 Q88=100.0*Q8/C(L)	00000705
146*	PRINT 551,X(L),Y(L),C(L),Q8,Q88	00000710
147*	GO TO 550	00000715
148*	548 PRINT 552,X(L),Y(L),C(L),Q8	00000720
149*	550 CONTINUE	00000725
150*	551 FORMAT(1P5E14.6)	00000730
151*	552 FORMAT(1P4E14.6,3X,'INFINITE')	
152*	555 CONTINUE	00000740
153*	T55=SQRT(T5)	00000745
154*	PRINT 553,T55	00000750
155*	553 FORMAT(/,10X,'STD ERROR OF ESTIMATE FOR Y = ',1PE14.6)	
156*	C	00000760
157*	C THIS IS THE END OF THE PART POLFT	00000765

158* C		00000770
159* C	TO ELIMINATE THE DIFFICULTIES ASSOCIATED WITH THE USE OF	00000775
160* C	LARGE NUMBERS THE INTERNAL CALCULATIONS OF THIS PROGRAM	00000780
161* C	ARE DONE IN THE UNITS LISTED AS FOLLOWS, BUT THE UNITS	00000785
162* C	OF THE INPUT DATA ARE IN INCHES	00000790
163* C		00000795
164* C	UNITS OF MASS - KIP-SEC**2/MILL	00000800
165* C	UNITS OF WEIGHT - KIP	00000805
166* C	UNITS OF DAMPING - KIP-SEC/MILL	00000810
167* C	UNITS OF DYNAMIC STIFFNESS - KIP/MILL	00000815
168* C	UNITS OF NONLINEAR COEFF. B - KIP/MILL**3	00000820
169* C	UNIT OF NONLINEAR COEFF. E -KIP/MILL**5	00000825
170* C		00000830
171* 789	CONTINUE	00000835
172*	AG=384000.0	00000840
173*	DSMM = 980	00000845
174*	FDYN=15.0	00000850
175*	CUR = 1.0	00000855
176*	CAS=DSMM/50.0	00000860
177*	CDS=DSMM/250.0	00000865
178*	CZS=DSMM/500.0	00000870
179*	AC1=2500.0	00000875
180*	AC2=6000.0	00000880
181*	AC3=9500.0	00000885
182*	IF(DSMM-3500)120,120,122	00000890
183* 120	CONTINUE	00000895
184*	C10=3.0E02	00000900
185*	C11=-4.3282600E02	00000905
186*	C12=3.2997113E02	00000910
187*	C13=-1.5834140E02	00000915
188*	C14=5.0866847E01	00000920
189*	C15=-1.1216281E01	00000925
190*	C16=1.7136733E00	00000930
191*	C17=-1.8075704E-01	00000935
192*	C18=1.2897555E-02	00000940
193*	C19=-5.9370854E-04	00000945
194*	C110=1.5899722E-05	00000950
195*	C111=-1.8806100E-07	00000955
196*	GOTO 126	00000960
197* 122	CONTINUE	00000965
198*	C10=1.3446749E04	00000970
199*	C11=-4.9537759E03	00000975
200*	C12=7.7413409E02	00000980
201*	C13=-6.6525920E01	00000985
202*	C14=3.3974423E00	00000990
203*	C15=-1.0316724E-01	00000995
204*	C16=1.7256088E-03	00010000
205*	C17=-1.2269488E-05	00010005
206*	C18=0.0	00010010
207*	C19=0.0	00010015
208*	C110=0.0	00010020
209*	C111=0.0	00010025
210* 126	CONTINUE	00010030

211*	C1A=C10+C11*CDS+C12*CDS**2+C13*CDS**3+C14*CDS**4+C15*CDS**5	00001035
212*	C1B=C16*CDS**6+C17*CDS**7+C18*CDS**8+C19*CDS**9	00001040
213*	C1C=C110*CDS**10+C111*CDS**11	00001045
214*	C1=C1A+C1B+C1C	00001050
215*	IF(DSMM-800.0)128,128,129	00001055
216*	128 CONTINUE	00001060
217*	CZ0=5.8102535E-07	00001065
218*	CZ1=-9.3288026E-04	00001070
219*	CZ2=2.5001039E-03	00001075
220*	CZ3=-2.3495371E-03	00001080
221*	CZ4=1.1630954E-03	00001085
222*	CZ5=-3.4300115E-04	00001090
223*	CZ6=6.3936609E-05	00001095
224*	CZ7=-7.7173709E-06	00001100
225*	CZ8=6.0093755E-07	00001105
226*	CZ9=-2.9097831E-08	00001110
227*	CZ10=7.9588319E-10	00001115
228*	CZ11=-9.3862036E-12	00001120
229*	GOTO 132	00001125
230*	129 CONTINUE	00001130
231*	IF(DSMM-3500)140,140,142	00001135
232*	140 CONTINUE	00001140
233*	CZ0=-5.2014956E00	00001145
234*	CZ1=1.8290391E00	00001150
235*	CZ2=-2.8868848E-01	00001155
236*	CZ3=2.6890909E-02	00001160
237*	CZ4=-1.6353769E-03	00001165
238*	CZ5=6.7947483E-05	00001170
239*	CZ6=-1.9633744E-06	00001175
240*	CZ7=3.9404340E-08	00001180
241*	CZ8=-5.3807386E-10	00001185
242*	CZ9=4.7621674E-12	00001190
243*	CZ10=-2.4601875E-14	00001195
244*	CZ11=5.6258519E-17	00001200
245*	GOTO 132	00001205
246*	142 CONTINUE	00001210
247*	CZ0=6.1111277E02	00001215
248*	CZ1=-4.9661811E01	00001220
249*	CZ2=1.7522450E00	00001225
250*	CZ3=-3.5075306E-02	00001230
251*	CZ4=4.3588585E-04	00001235
252*	CZ5=-3.4449323E-06	00001240
253*	CZ6=1.6915632E-08	00001245
254*	CZ7=-4.7199626E-11	00001250
255*	CZ8=5.7320258E-14	00001255
256*	CZ9=0.0	00001260
257*	CZ10=0.0	00001265
258*	CZ11=0.0	00001270
259*	132 CONTINUE	00001275
260*	CC1A=CZ0+CZ1*CAS+CZ2*CAS**2+CZ3*CAS**3+CZ4*CAS**4+CZ5*CAS**5	00001280
261*	CC1B=CZ6*CAS**6+CZ7*CAS**7+CZ8*CAS**8+CZ9*CAS**9+CZ10*CAS**10	00001285
262*	CC1C=CZ11*CAS**11	00001290
263*	CC1=1000.0*(CC1A+CC1B+CC1C)	00001295

264*	BC2=0.02	00001300
265*	DEL=0.001	00001305
266*	BC1=15.0	00001310
267*	IF(DSMM-3500)144,144,146	00001315
268*	144 CONTINUE	00001320
269*	CD0=-7.9887704E-05	00001325
270*	CD1=6.3085251E-04	00001330
271*	CD2=-7.4214518E-06	00001335
272*	CD3=-1.1141749E-05	00001340
273*	CD4=2.3517785E-06	00001345
274*	CD5=-2.0060899E-07	00001350
275*	CD6=9.4187498E-09	00001355
276*	CD7=-2.6952086E-10	00001360
277*	CD8=4.8360439E-12	00001365
278*	CD9=-5.3232761E-14	00001370
279*	CD10=3.2890826E-16	00001375
280*	CD11=-8.7376943E-19	00001380
281*	GOTO 148	00001385
282*	146 CONTINUE	00001390
283*	CD0=-1.7802618E01	00001395
284*	CD1=1.2909806E00	00001400
285*	CD2=-3.9798780E-02	00001405
286*	CD3=6.7678723E-04	00001410
287*	CD4=-6.8501186E-06	00001415
288*	CD5=4.1241283E-08	00001420
289*	CD6=-1.3668450E-10	00001425
290*	CD7=1.9232534E-13	00001430
291*	CD8=0.0	00001435
292*	CD9=0.0	00001440
293*	CD10=0.0	00001445
294*	CD11=0.0	00001450
295*	148 CONTINUE	00001455
296*	C01=CD0+CD1+CAS+CD2+CAS**2+CD3+CAS**3+CD4+CAS**4+CD5+CAS**5	00001460
297*	C02=CD6+CAS**6+CD7+CAS**7+CD8+CAS**8+CD9+CAS**9+CD10+CAS**10	00001465
298*	C03=CD11+CAS**11	00001470
299*	C0=C01+C02+C03	00001475
300*	IF(DSMM-3500)150,150,152	00001480
301*	150 CONTINUE	00001485
302*	C20=1.7886330E-01	00001490
303*	C21=-6.4582961E-03	00001495
304*	C22=-1.4994404E-03	00001500
305*	C23=1.9047721E-04	00001505
306*	C24=-5.7220492E-06	00001510
307*	C25=-2.8715333E-07	00001515
308*	C26=2.8929761E-08	00001520
309*	C27=-1.0573124E-09	00001525
310*	C28=2.1306976E-11	00001530
311*	C29=-2.4954492E-13	00001535
312*	C210=1.5955587E-15	00001540
313*	C211=-4.3182053E-18	00001545
314*	GOTO 156	00001550
315*	152 CONTINUE	00001555
316*	C20=1.8711536E03	00001560

317*	C21=-1.3477721E02	00001565
318*	C22=4.1114309E00	00001570
319*	C23=-6.8902750E-02	00001575
320*	C24=6.8561083E-04	00001580
321*	C25=-4.0530542E-06	00001585
322*	C26=1.3187613E-08	00001590
323*	C27=-1.8228129E-11	00001595
324*	C28=0.0000000000000000	00001600
325*	C29=0.0000000000000000	00001605
326*	C210=0.0000000000000000	00001610
327*	C211=0.0000000000000000	00001615
328*	156 CONTINUE	00001620
329*	C2A=C20+C21*CAS+C22*CAS**2+C23*CAS**3+C24*CAS**4+C25*CAS**5	00001625
330*	C2B=C26*CAS**6+C27*CAS**7+C28*CAS**8+C29*CAS**9	00001630
331*	C2C=C210*CAS**10+C211*CAS**11	00001635
332*	C2=1000.0*(C2A+C2B+C2C)	00001640
333*	IF(DSMM-3500)134,134,136	00001645
334*	134 CONTINUE	00001650
335*	CK0=-7.8699233E-05	00001655
336*	CK1=3.2234471E-01	00001660
337*	CK2=1.4716139E00	00001665
338*	CK3=-1.1769343E00	00001670
339*	CK4=4.9592895E-01	00001675
340*	CK5=-1.3306910E-01	00001680
341*	CK6=2.3581299E-02	00001685
342*	CK7=-2.7710718E-03	00001690
343*	CK8=2.1279022E-04	00001695
344*	CK9=-1.0238109E-05	00001700
345*	CK10=2.7966863E-07	00001705
346*	CK11=-3.3061349E-09	00001710
347*	GOTO 138	00001715
348*	136 CONTINUE	00001720
349*	CK0=-3.1842085E05	00001725
350*	CK1=1.2488155E05	00001730
351*	CK2=-2.0808714E04	00001735
352*	CK3=1.9071617E03	00001740
353*	CK4=-1.0375390E02	00001745
354*	CK5=3.3493295E00	00001750
355*	CK6=-5.9403607E-02	00001755
356*	CK7=4.4664367E-04	00001760
357*	CK8=0.0	00001765
358*	CK9=0.0	00001770
359*	CK10=0.0	00001775
360*	CK11=0.0	00001780
361*	138 CONTINUE	00001785
362*	CKAP1=CK0+CK1*CDS+CK2*CDS**2+CK3*CDS**3+CK4*CDS**4+CK5*CDS**5	00001790
363*	CKAP2=CK6*CDS**6+CK7*CDS**7+CK8*CDS**8+CK9*CDS**9+CK10*CDS**10	00001795
364*	CKAP3=CK11*CDS**11	00001800
365*	CKAP=CKAP1+CKAP2+CKAP3	00001805
366*	CC2=(1.0+CKAP+CKAP**2)/3.0	00001810
367*	EOB=1.3051737	00001815
368*	COB=0.00698505	00001820
369*	EOA=1.1234066	00001825

370*	COA=0.0103823	00001830
371*	FCT1=1.0	00001835
372*	FCT2=1.0	00001840
373*	FCT3=1.0	00001845
374*	FCT4=1.0	00001850
375*	FREQ=15.0	00001855
376*	FREQT=15.0	00001860
377*	FREQR=8.0	00001865
378*	IF(DSMM-3500)160,162,162	00001870
379*	160 CONTINUE	00001875
380*	BBF0=-5.2502975E-02	00001880
381*	BBF1=2.4290747E-01	00001885
382*	BBF2=-5.3572645E-02	00001890
383*	BBF3=6.6367522E-03	00001895
384*	BBF4=-2.850117E-04	00001900
385*	BBF5=-4.2239845E-06	00001905
386*	BBF6=8.6076538E-07	00001910
387*	BBF7=-3.5906169E-08	00001915
388*	BBF8=7.7332822E-10	00001920
389*	BBF9=-9.4366145E-12	00001925
390*	BBF10=6.2002737E-14	00001930
391*	BBF11=-1.7094962E-16	00001935
392*	GOTO 164	00001940
393*	162 CONTINUE	00001945
394*	BBF0=-1.6156112E02	00001950
395*	BBF1=1.2157621E01	00001955
396*	BBF2=-3.7874985E-01	00001960
397*	BBF3=6.3966447E-03	00001965
398*	BBF4=-6.3479456E-05	00001970
399*	BBF5=3.7107308E-07	00001975
400*	BBF6=-1.1851895E-09	00001980
401*	BBF7=1.5976850E-12	00001985
402*	BBF8=0.0	00001990
403*	BBF9=0.0	00001995
404*	BBF10=0.0	00002000
405*	BBF11=0.0	00002005
406*	164 CONTINUE	00002010
407*	BBFA=BBF0+BBF1*CAS+BBF2*CAS**2+BBF3*CAS**3+BBF4*CAS**4	00002015
408*	BBFB=BBF5*CAS**5+BBF6*CAS**6+BBF7*CAS**7+BBF8*CAS**8	00002020
409*	BBFC=BBF9*CAS**9+BBF10*CAS**10+BBF11*CAS**11	00002025
410*	BBF=BBFA+BBFB+BBFC	00002030
411*	IF(DSMM-3500)180,180,182	00002035
412*	180 CONTINUE	00002040
413*	EEF0=-1.6392956E-04	00002045
414*	EEF1=3.8696350E-02	00002050
415*	EEF2=-7.5764972E-03	00002055
416*	EEF3=9.1875738E-04	00002060
417*	EEF4=-7.1071983E-05	00002065
418*	EEF5=3.6245638E-06	00002070
419*	EEF6=-1.2456023E-07	00002075
420*	EEF7=2.8953761E-09	00002080
421*	EEF8=-4.4780369E-11	00002085
422*	EEF9=4.4053809E-13	00002090

423*	EEF10=-2.4905921E-15	00002095
424*	EEF11=6.1533305E-18	00002100
425*	GOTO 189	00002105
426*	182 CONTINUE	00002110
427*	IF(DSM-4000)184,184,186	00002115
428*	184 CONTINUE	00002120
429*	EEF0=2.9869271E04	00002125
430*	EEF1=-1.9805576E03	00002130
431*	EEF2=5.2478918E01	00002135
432*	EEF3=-6.9458127E-01	00002140
433*	EEF4=4.5919943E-03	00002145
434*	EEF5=-1.2131410E-05	00002150
435*	EEF6=0.0	00002155
436*	EEF7=0.0	00002160
437*	EEF8=0.0	00002165
438*	EEF9=0.0	00002170
439*	EEF10=0.0	00002175
440*	EEF11=0.0	00002180
441*	GOTO 189	00002185
442*	186 CONTINUE	00002190
443*	EEF0=1.588114E02	00002195
444*	EEF1=-1.0509675E01	00002200
445*	EEF2=2.9687507E-01	00002205
446*	EEF3=-4.6404242E-03	00002210
447*	EEF4=4.3351686E-05	00002215
448*	EEF5=-2.4208222E-07	00002220
449*	EEF6=7.4825315E-10	00002225
450*	EEF7=-9.8764637E-13	00002230
451*	EEF8=0.0	00002235
452*	EEF9=0.0	00002240
453*	EEF10=0.0	00002245
454*	EEF11=0.0	00002250
455*	189 CONTINUE	00002255
456*	EEFA=EEF0+EEF1*CAS+EEF2*CAS**2+EEF3*CAS**3+EEF4*CAS**4	00002260
457*	EEFB=EEF5*CAS**5+EEF6*CAS**6+EEF7*CAS**7+EEF8*CAS**8	00002265
458*	EEFC=EEF9*CAS**9+EEF10*CAS**10+EEF11*CAS**11	00002270
459*	EEF=EEFA+EEFB+EEFC	00002275
460*	FS=16.0	00002280
461*	PI=3.14159265	00002285
462*	POIS1=0.3	00002290
463*	POIS2 = 0.3	00002295
464*	POIS3 = 0.35	00002300
465*	POIS4 = 0.35	00002305
466*	OMEGT=2.0*PI*FREQT	00002310
467*	OMEGR=2.0*PI*FREQR	00002315
468*	EMD1 = 1.3*10**6	00002320
469*	EMD2 = 1.3*10**6	00002325
470*	EMD3 = 4.0*10**4	00002330
471*	EMD4 = 10000	00002335
472*	633 EMD4 = EMD4+1000	00002340
473*	GMD1=EMD1/(2.0*(1.0+POIS1))	00002345
474*	GMD2=EMD2/(2.0*(1.0+POIS2))	00002350
475*	GMD3=EMD3/(2.0*(1.0+POIS3))	00002355

AD-A087 186

ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG--ETC F/G 1/5
PAVEMENT EVALUATION AND OVERLAY DESIGN USING VIBRATORY NONDESTR--ETC(U)
MAR 80 R A WEISS DOT-FA73WAI-377

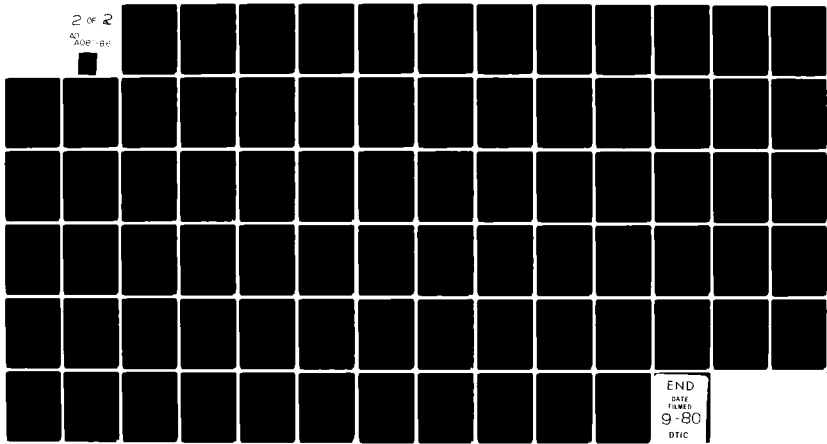
UNCLASSIFIED

FAA-RD-77-186-VOL-1

NL

2 of 2

AD-A087 186



END
DATE
FILMED
9-80
DTIC

476*	GMD4=EMD4/(2.0*(1.0+POIS4))	00002360
477*	EMOD1=EMD1/1.0E09	00002365
478*	EMOD2=EMD2/1.0E09	00002370
479*	EMOD3=EMD3/1.0E09	00002375
480*	EMOD4=EMD4/1.0E09	00002380
481*	SHEAR1=GMD1/1.0E09	00002385
482*	SHEAR2=GMD2/1.0E09	00002390
483*	SHEAR3=GMD3/1.0E09	00002395
484*	SHEAR4=GMD4/1.0E09	00002400
485*	Q1=(1.0-POIS1)/(1.0-2.0*POIS1)	00002405
486*	Q2=(1.0-POIS2)/(1.0-2.0*POIS2)	00002410
487*	Q3=(1.0-POIS3)/(1.0-2.0*POIS3)	00002415
488*	Q4=(1.0-POIS4)/(1.0-2.0*POIS4)	00002420
489*	S1=(1.0-POIS1)*Q1	00002425
490*	S2=(1.0-POIS2)*Q2	00002430
491*	S3=(1.0-POIS3)*Q3	00002435
492*	S4=(1.0-POIS4)*Q4	00002440
493*	H1=CC2*PI*S1*AC1/2.0	00002445
494*	H2=CC2*PI*S2*(AC2-AC1)/2.0	00002450
495*	H3=CC2*PI*S3*(AC3-AC2)/2.0	00002455
496*	H1 = 3.0	00002460
497*	H2 = 3.0	00002465
498*	H3 = 7.0	00002470
499*	HT1=H1	00002475
500*	HT2=H2	00002480
501*	HT3=H3	00002485
502*	H1=H1*1000	00002490
503*	H2=H2*1000	00002495
504*	H3=H3*1000	00002500
505*	HH2=H1+H2	00002505
506*	HH3=H1+H2+H3	00002510
507*	ARG=H1*EMOD1+H2*EMOD2+H3*EMOD3+EMOD4	00002515
508*	715 FORMAT(/,1H ,6X,2HA1,12X,2HA2,12X,2HA3,12X,2HA4/1P4E14.6)	00002520
509*	OMEG=2.0*PI*FREQ	00002525
510*	SO=1.0/U(2)	00002530
511*	ALPH1=U(4)*SO**5	00002535
512*	ALPH1=ALPH1/1.414	00002540
513*	ALPH2=U(6)*SO**9	00002545
514*	ALPH2=ALPH2/1.6378	00002550
515*	W=16.0*CC1	00002555
516*	VM=W/AG	00002560
517*	655 FORMAT(/, ' WE,WR,FREQT',/,1P3E14.6)	
518*	SKO=VM*OMEG**2+SQRT((SO**2)-(CO*OMEG)**2)	00002570
519*	IF(FREQ-16.0)227,5,5	
520*	5 SKO=VM*OMEG**2-SQRT((SO**2)-(CO*OMEG)**2)	00002580
521*	227 SK1=VM*OMEG**2	00002585
522*	SK2=SKO-SK1	00002590
523*	OMEGC=SQRT(SK0/VM)	00002595
524*	FREQC=OMEGC/(2.0*PI)	00002600
525*	FPSI=(FDYN**2)/SO**4	00002605
526*	ADYN=(FDYN/SO)*(1.0+ALPH1*FPSI+ALPH2*FPSI**2)	00002610
527*	BB=-ALPH1/(C1*(SKO-VM*OMEG**2))	00002615
528*	BB=-ABS(BB)	00002620

529*	ZZ=7./2.*(C1**2)*(BB**2)*((SKO-VM*OMEG**2)**2)-ALPH2	00002625
530*	WM=C2*(SKO-VM*OMEG**2)	00002630
531*	VV=(C1**2)*(BB**2)/(2.0*C2*(SKO-VM*OMEG**2))	00002635
532*	EE=(ZZ/WM)-VV	00002640
533*	EE=ABS(EE)	00002645
534*	SK=SKO+C1*BB*BBF*(ADYN**2)+C2*EE*EEF*(ADYN**4)	00002650
535*	STIF=SQRT((SK-VM*OMEG**2)**2+(C0*OMEG)**2)	00002655
536*	PS=FS/(PI*AO**2)	00002660
537*	POIS=POIS1	00002665
538*	QO=Q1	00002670
539*	SHEAR=SHEAR1	00002675
540*	DLO=PI*AO*((1.0-POIS)**2)/(2.0*(1.0-2.0*POIS))	00002680
541*	DLO=DLO*CC2	00002685
542*	IF(DLO-H1)6,6,210	
543*	6 FK00=2.0*PI*(AO**2)*Q1*SHEAR1/DLO	00002695
544*	FK00=FK00*CC2	00002700
545*	DL2=-BB*(DLO**2)/(4.0*PI*(AO**2)*QO*SHEAR)	00002705
546*	DL2=DL2/CC2	00002710
547*	CL4=DLO*(DL2/DLO)**2	00002715
548*	EL4=EE*(DLO**2)/(6.0*PI*(AO**2)*QO*SHEAR)	00002720
549*	EL4=EL4/CC2	00002725
550*	DL4=CL4-EL4	00002730
551*	BBA=-4.0*PI*(AO**2)*Q1*SHEAR1*DL2/(DLO**2)	00002735
552*	BBA=BBA*CC2	00002740
553*	DELTA=(DL2/DLO)**2-DL4/DLO	00002745
554*	EEA=6.0*PI*(AO**2)*Q1*SHEAR1*DELTA/DLO	00002750
555*	EEA=EEA*CC2	00002755
556*	GOTO220	00002760
557*	210 DLO=CC2*(S2*AO+AC1*(S1-S2))*PI/2.0	00002765
558*	IF(DLO-HH2)7,7,230	
559*	7 FLO=2.0*PI*(AO**2)/DLO**2	00002775
560*	FK00=FLO*(H1*(Q1*SHEAR1-Q2*SHEAR2)+Q2*SHEAR2*DLO)	00002780
561*	FK00=FK00*CC2	00002785
562*	FL2=2.0*H1*(Q1*SHEAR1-Q2*SHEAR2)+Q2*SHEAR2*DLO	00002790
563*	DL2=-BB*(DLO**3)/(4.0*PI*(AO**2)*FL2)	00002795
564*	DL2=DL2/CC2	00002800
565*	FL4=3.0*H1*(Q1*SHEAR1-Q2*SHEAR2)+Q2*SHEAR2*DLO	00002805
566*	DL4=((DL2**2)*FL4/DLO-EE*(DLO**3)/(6.0*CC2*PI*AO**2))/FL2	00002810
567*	QG12=Q1*SHEAR1-Q2*SHEAR2	00002815
568*	BBA=-4.0*PI*(AO**2)*DL2*(2.0*H1*QG12+Q2*SHEAR2*DLO)/(DLO**3)	00002820
569*	BBA=BBA*CC2	00002825
570*	RHO=3.0*((DL2/DLO)**2)-2.0*DL4/DLO	00002830
571*	DELTA=(DL2/DLO)**2-DL4/DLO	00002835
572*	EEA=6.0*PI*(AO**2)*(RHO*H1*QG12+DELTA*Q2*SHEAR2*DLO)/(DLO**2)	00002840
573*	EEA=EEA*CC2	00002845
574*	GOTO220	00002850
575*	230 DLO=CC2*(S3*AO+AC2*(S2-S3)+AC1*(S1-S2))*PI/2.0	00002855
576*	IF(DLO-HH3)8,8,240	
577*	8 QG13=Q1*SHEAR1-Q3*SHEAR3	00002865
578*	QG23=Q2*SHEAR2-Q3*SHEAR3	00002870
579*	FLO=2.0*PI*(AO**2)/DLO**2	00002875
580*	FK00=CC2*FLO*(H1*QG13+H2*QG23+Q3*SHEAR3*DLO)	00002880
581*	GL2=2.0*(H1*QG13+H2*QG23)+Q3*SHEAR3*DLO	00002885

582*	GL22=GL2/2.0	00002890
583*	DL2=-BB*(DLO**3)/(4.0*PI*(AO**2)*CC2*GL2)	00002895
584*	FL4=3.0*(H1*QG13+H2*QG23)+Q3*SHEAR3*DLO	00002900
585*	DL4=((DL2**2)*FL4/DLO-EE*(DLO**3)/(6.0*PI*CC2*AO**2))/GL2	00002905
586*	BBA=-4.0*PI*(AO**2)*DL2*CC2*GL2/DLO**3	00002910
587*	RHO=3.0*((DL2/DLO)**2)-2.0*DL4/DLO	00002915
588*	DELTA=(DL2/DLO)**2-DL4/DLO	00002920
589*	EEA1=6.0*PI*(AO**2)*CC2/DLO**2	00002925
590*	EEA=EEA1*(RHO*(H1*QG13+H2*QG23)+DELTA*Q3*SHEAR3*DLO)	00002930
591*	GOTO220	00002935
592*	240 DLO=CC2*(S4*AO+AC3*(S3-S4)+AC2*(S2-S3)+AC1*(S1-S2))*PI/2.0	00002940
593*	DLO0=-1.6182520E-01	00002945
594*	DLO1=2.9406204E01	00002950
595*	DLO2=-1.5550641E01	00002955
596*	DLO3=1.1928522E01	00002960
597*	DLO4=-4.4597082E00	00002965
598*	DLO5=8.8599714E-01	00002970
599*	DLO6=-1.0401599E-01	00002975
600*	DLO7=7.6323389E-03	00002980
601*	DLO8=-3.5492892E-04	00002985
602*	DLO9=1.0179406E-05	00002990
603*	DLO10=-1.6440288E-07	00002995
604*	DLO11=1.1443959E-09	00003000
605*	DLOA=DLO0+DLO1*CDS+DLO2*CDS**2+DLO3*CDS**3+DLO4*CDS**4	00003005
606*	DLOB=DLO5*CDS**5+DLO6*CDS**6+DLO7*CDS**7+DLO8*CDS**8	00003010
607*	DLOC=DLO9*CDS**9+DLO10*CDS**10+DLO11*CDS**11	00003015
608*	DLO=1000.0*(DLOA+DLOB+DLOC)	00003020
609*	ALPP=(CKAP-1.0)*AO/DLO	00003025
610*	FAD0=1.0+ALPP*DLO/AO+((ALPP*DLO/AO)**2)/3.0	00003030
611*	FAN1=1.0+ALPP*H1/AO+((ALPP*H1/AO)**2)/3.0	00003035
612*	FAB2=H1*HH2+H1**2+HH2**2	00003040
613*	FAN2=1.0+ALPP*(H1+HH2)/AO+((ALPP/AO)**2)*FAB2/3.0	00003045
614*	FAB3=HH2*HH3+HH2**2+HH3**2	00003050
615*	FAN3=1.0+ALPP*(HH2+HH3)/AO+((ALPP/AO)**2)*FAB3/3.0	00003055
616*	FAB4=HH3*DLO+HH3**2+DLO**2	00003060
617*	FAN4=1.0+ALPP*(HH3+DLO)/AO+((ALPP/AO)**2)*FAB4/3.0	00003065
618*	FAC1=FAN1/FAD0	00003070
619*	FAC2=FAN2/FAD0	00003075
620*	FAC3=FAN3/FAD0	00003080
621*	FAC4=FAN4/FAD0	00003085
622*	QG14=Q1*SHEAR1-Q4*SHEAR4	00003090
623*	QG24=Q2*SHEAR2-Q4*SHEAR4	00003095
624*	QG34=Q3*SHEAR3-Q4*SHEAR4	00003100
625*	FLO=2.0*PI*(AO**2)/DLO**2	00003105
626*	FK00=CC2*FLO*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)	00003110
627*	FK00=FK00+Q4*SHEAR4*DLO*FAC4	00003115
628*	GL2=2.0*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)	00003120
629*	GL2=GL2+Q4*SHEAR4*DLO*FAC4	00003125
630*	DL2=-BB*(DLO**3)/(4.0*PI*(AO**2)*CC2*GL2)	00003130
631*	FL4=3.0*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)	00003135
632*	FL4=FL4+Q4*SHEAR4*DLO*FAC4	00003140
633*	DL4=((DL2**2)*FL4/DLO-EE*(DLO**3)/(6.0*PI*CC2*AO**2))/GL2	00003145
634*	BBA=-4.0*PI*(AO**2)*DL2*CC2*GL2/DLO**3	00003150

```

635*      RHO=3.0*((DL2/DLO)**2)-2.0*DL4/DLO      00003155
636*      DELTA=(DL2/DLO)**2-DL4/DLO      00003160
637*      EEA1=6.0*PI*(A0**2)*CC2/DLO**2      00003165
638*      EEA2=RHO*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)      00003170
639*      EEA2=EEA2+DELTA*Q4*SHEAR4*DLO*FAC4      00003175
640*      EEA=EEA1*EEA2      00003180
641*      220 CONTINUE      00003185
642*      ARG2=H1*EMOD1+H2*EMOD2+H3*EMOD3+(DLO-HH3)*EMOD4      00003190
643* C      00003195
644* C THE FOLLOWING GIVES THE CALCULATION OF THE STATIC ELASTIC      00003200
645* C DISPLACEMENT OF THE PAVEMENT SURFACE      00003205
646* C      00003210
647*      ZP(1)=-1.0*FS      00003215
648*      ZP(2)=FK00      00003220
649*      ZP(3)=0.0      00003225
650*      ZP(4)=BBA      00003230
651*      ZP(5)=0.0      00003235
652*      ZP(6)=EEA      00003240
653*      CALLDOWNH(ZP,5,RR,CR)      00003245
654*      680 FORMAT(//5(4HROOT,I2,5X,1P2E20.7//)      00003250
655*      & 5X,3H-FS,5X,1PE20.7/5X,4HFK00,4X,1PE20.7/13X,1PE20.7/      00003255
656*      & 5X,3HBBA,5X,1PE20.7/13X,1PE20.7/5X,3HEEA,5X,1PE20.7/      00003260
657*      XE=0.0      00003265
658*      D0221K=1,5      00003270
659*      6'      00003285
660*      15 IF(RR(K)-XE)221,221,16      00003290
661*      16 XE=RR(K)      00003295
662* C      00003300
663*      221 CONTINUE      00003305
664*      VME=VM+BC2*SK0*((BC1-FREQT)/OMEGT)**2      00003310
665*      WE=CC1      00003315
666*      VMR=VM+BC2*SK0*((BC1-FREQR)/OMEGR)**2      00003320
667*      WR=VMR*AG/16.0      00003325
668*      OMEG=OMEGT      00003330
669*      RBA=4.0      00003335
670*      REA=(1.0-CUR)*180.0+CUR*17.0      00003340
671*      THETA=(4.0/3.0)*C1*EBF      00003345
672*      ETA=(8.0/5.0)*C2*EEF      00003350
673*      SKOT=FK00+3.0*RBA*BBA*(XE**2)+5.0*REA*EEA*(XE**4)      00003355
674*      CCC=CO      00003360
675*      SOT=SQRT((SKOT-VM*OMEG**2)**2+(CCC*OMEG)**2)      00003365
676*      SKT=SKOT+C1*BBA*BBF*(ADYN**2)+C2*EEA*EEF*(ADYN**4)      00003370
677*      STIFT=SQRT((SKT-VM*OMEG**2)**2+(CCC*OMEG)**2)      00003375
678*      FDYNT=FDYN      00003380
679*      FPSIT=(FDYNT**2)/SOT**4      00003385
680*      ADYNT=(FDYNT/SOT)*(1.0+ALPH1*FPSIT+ALPH2*FPSIT**2)      00003390
681*      OMETR=SQRT(SKOT/VMR)      00003395
682*      OM=(C1*BBF*BBA*(ADYN**2)+C2*EEF*EEA*(ADYN**4))/VMR      00003400
683*      OMEGTR=SQRT(OMETR**2+OM-2.0*(CO/(2.0*VMR))**2)      00003405
684*      FREQTR=OMEGTR/(2.0*PI)      00003410
685*      VMET=VM+BC2*SKOT*((BC1-FREQT)/OMEGT)**2      00003415
686*      WET=VMET*AG/16.0      00003415
687*      VMRT=VM+BC2*SKOT*((BC1-FREQTR)/OMEGTR)**2

```

```

688* WRT=VMRT*AG/16.0 00003420
689* SOTT=SQRT((SKOT-VMET*OMEGT**2)**2+(CCC*OMEGT)**2) 00003425
690* ALPHT1=-BBA*C1*(SKOT-VMET*OMEGT**2) 00003430
691* WW=C2*(SKOT-VMET*OMEGT**2) 00003435
692* VV=(C1**2)*(BBA**2)/(2.0*C2*(SKOT-VMET*OMEGT**2)) 00003440
693* ZZ=(EEA+VV)*WW 00003445
694* ALPHT2=7./2.*(C1**2)*(BBA**2)*((SKOT-VMET*OMEGT**2)**2)-ZZ 00003450
695* FPSITT=(FDYNT**2)/SOTT**4 00003455
696* ADYNTT=(FDYNT/SOTT)*(1.0+ALPHT1*FPSITT+ALPHT2*FPSITT**2) 00003460
697* SKTT=SKOT+C1*BBF*BBA*(ADYNT**2)+C2*EEF*EEA*(ADYNTT**4) 00003465
698* STIFTT=SQRT((SKTT-VMET*OMEGT**2)**2+(CCC*OMEGT)**2) 00003470
699* DEAN=FDYNT/ADYNTT
700* DFDYN=FDYNT+DEL 00003480
701* DFPIST=(DFDYN**2)/SOTT**4 00003485
702* DADYN=(DFDYN/SOTT)*(1.0+ALPHT1*DFPIST+ALPHT2*DFPIST**2) 00003490
703* DSM=DEL/(DADYN-ADYNTT) 00003495
704* DSMT=1000.0*DSM 00003500
705* DELDSM=DSM-DSMT 00003505
706* IF(DSMH-DSMT)633,733,733 00003510
707* 733 CONTINUE 00003515
708* STIF=STIF*1000. 00003520
709* STIFT=STIFT*1000. 00003525
710* WRITE(6,40)
711* 40 FORMAT(' STIF STIFT DSMH DSM',
712* &' T')
713* PRINT551,STIF,STIFT,DSMH,DSMT 00003540
714* WRITE(6,41)
715* 41 FORMAT(' EMD1 EMD2 EMD3 EMD4') 00003550
716* PRINT551,EMD1,EMD2,EMD3,EMD4
717* WRITE(6,42)
718* 42 FORMAT(' GMD1 GMD2 GMD3 GMD4') 00003560
719* PRINT551,GMD1,GMD2,GMD3,GMD4
720* WRITE(6,43)
721* 43 FORMAT(' POIS1 POIS2 POIS3 POIS4') 00003570
722* PRINT551,POIS1,POIS2,POIS3,POIS4
723* WRITE(6,44)
724* 44 FORMAT(' HT1 HT2 HT3')
725* PRINT551,HT1,HT2,HT3 00003580
726* C 00003585
727* C THIS PART GIVES A SECOND METHOD OF CALCULATING THE STATIC 00003590
728* C ELASTIC DISPLACEMENT OF THE PAVEMENT SURFACE 00003595
729* C 00003600
730* ZP(2)=SKO 00003605
731* ZP(4)=-2.0*BB 00003610
732* ZP(5)=-10.0*FS/SO*EE 00003615
733* ZP(6)=-4.0*EE 00003620
734* CALLDOWNH(ZP,5,RR,CR) 00003625
735* XEE=0.0 00003630
736* DO223K=1,5 00003635
737* IF(CR(K))223,19,223
738* 19 IF(XEE-RR(K))21,223,223
739* 21 IF(RR(K))223,223,22
740* 22 XEE=RR(K)

```

741*	C		00003660
742*	223	CONTINUE	00003665
743*		SK00=FS/XEE-BB*(XEE**2)-EE*(XEE**4)	00003670
744*		STOP	00003680
745*	570	WRITE(6,45)	
746*	45	FORMAT('PROGRAM SIZE LIMIT IS 100 DATA POINTS.')	
747*		GO TO 628	00003690
748*	576	WRITE(6,46)	
749*	46	FORMAT('ELEVENTH DEGREE IS THE LIMIT.')	
750*	592	FORMAT(F6,0)	00003750
751*	590	READ(FNAME,592)AO	00003755
752*		N=5	00003765
753*		X(1)=0.0	
754*		Y(1)=0.0	
755*		DO6001=2,98,2	00003775
756*	593	FORMAT(F4.0,1X,F6.2)	00003780
757*		READ(FNAME,593,END=602)X(I),Y(I)	00003785
758*		X(I+1)=-X(I)	00003795
759*		Y(I+1)=-Y(I)	00003800
760*	600	CONTINUE	00003805
761*	602	M=I-1	00003810
762*		GO TO 36	00003815
763*	616	ITEMP=N-1	
764*		PRINT 617,ITEMP	
765*	617	FORMAT('/ TOO FEW POINTS FOR FITTING DEGREE',I4)	
766*	628	WRITE(6,47)	
767*	47	FORMAT('STOP')	
768*		STOP	
769*		END	
770*	C	SUBPROGRAM ZORP	00003840
771*	C		00003845
772*	C		00003850
773*	C	THE SUBPROGRAM ZORP2 CALCULATES THE ROOTS OF THE FIFTH ORDER	00003855
774*	C	POLYNOMIAL THAT CONNECTS THE STATIC LOAD AND THE STATIC	00003860
775*	C	ELASTIC DISPLACEMENT OF THE PAVEMENT SURFACE DIRECTLY BENEATH	00003865
776*	C	THE VIBRATOR BASEPLATE	00003870
777*	C	ZORP2	00003875
778*	C	ROUTINES FOR SOLVING POLYNOMIALS	00003880
779*		SUBROUTINE POLY(N,A,R,C,PR,PC,RHO,PHI)	00003885
780*		DIMENSION A(9999)	00003890
781*		IF(RHO)10,5,10	00003895
782*	5	R=A(1)	00003900
783*		C=0.	00003905
784*		PR=A(2)	00003910
785*		PC=0.	00003915
786*		RETURN	00003920
787*	10	V1=1.	00003925
788*		V2=0.	00003930
789*		R=A(1)	00003935
790*		C=0.	00003940
791*		PR=0.	00003945
792*		PC=0.	00003950
793*		W1=RHO*COS(PHI)	00003955

794*		W2=RHO*SIN(PHI)	00003960
795*		NN=N+1	00003965
796*		DO 20 I=2,NN	00003970
797*		T1=W1*V1-W2*V2	00003975
798*		V2=W2*V1+W1*V2	00003980
799*		V1=T1	00003985
800*		R=R+A(I)*V1	00003990
801*		C=C+A(I)*V2	00003995
802*		PR=PR+A(I)*(I-1)*V1	00004000
803*	20	PC=PC+A(I)*(I-1)*V2	00004005
804*		PR=PR/RHO	00004010
805*		PC=PC/RHO	00004015
806*	5001	RETURN	00004020
807*		END	00004025
808*		SUBROUTINE ARCTA(X,Y,ANGLE)	00004030
809*		PI=3.14159265	00004035
810*		IF(X)10,30,20	00004040
811*	10	ANGLE=ATAN(Y/X)+PI*SIGN(1.,Y)	00004045
812*		RETURN	00004050
813*	20	ANGLE=ATAN(Y/X)	00004055
814*		RETURN	00004060
815*	30	IF(Y)40,60,50	00004065
816*	40	ANGLE=-PI/2.	00004070
817*		RETURN	00004075
818*	50	ANGLE=PI/2.	00004080
819*		RETURN	00004085
820*	60	ANGLE=0.	00004090
821*		RETURN	00004095
822*		END	00004100
823*		SUBROUTINE DOWNH(A,NAR,RR,CR)	00004105
824*		DIMENSION A(9999),RR(9999),CR(9999),Q(101),B(3)	00004110
825*		J=0	00004115
826*		N=NAR	00004120
827*		NPL1=N+1	00004125
828*		ANPP=A(NPL1)	00004130
829*		DO 102 I=1,NPL1	00004135
830*		IF (A(I))103,102,103	00004140
831*	102	CONTINUE	00004145
832*	103	C=ABS(A(I)/A(NPL1))	00004150
833*		LU=120	00004155
834*		LL=-120	00004160
835*		IF(C-2.*LU)100,100,101	00004165
836*	100	IF(C-2.*LL)101,105,105	00004170
837*	101	NAR=-NAR	00004175
838*		GO TO 5001	00004180
839*	105	II=(LU+LL)/2	00004185
840*		IF(C-2.*II)110,110,109	00004190
841*	109	LL=II	00004195
842*		GO TO 111	00004200
843*	110	LU=II	00004205
844*	111	IF(LU-LL-1)5001,112,105	00004210
845*	112	IB=II/N	00004215
846*		IF(IB)114,120,114	00004220

847*	114	DO 115 I=1,NPL1	00004225
848*		II=I-1	00004230
849*	115	A(I)=A(I)*(2.*(II*IB))	00004235
850*	120	DO 121 J1=1,NPL1	00004240
851*	121	A(J1)=A(J1)/A(NPL1)	00004245
852*	201	IF(N)2001,2001,206	00004250
853*	206	IF(A(1))301,211,301	00004255
854*	211	J=J+1	00004260
855*		RR(J)=0.	00004265
856*		CR(J)=0.	00004270
857*		DO 221 J1=1,N	00004275
858*	221	A(J1)=A(J1+1)	00004280
859*		N=N-1	00004285
860*		GO TO 201	00004290
861*	301	IF(N-2)601,501,401	00004295
862*	401	CALL GRAD(A,N,X,Y)	00004300
863*	421	IF(ABS(Y)-ABS(X*1.E-4))431,431,441	00004305
864*	431	Y=0.	00004310
865*	441	J=J+1	00004315
866*		RR(J)=X	00004320
867*		CR(J)=Y	00004325
868*		IF(Y)461,1021,461	00004330
869*	461	J=J+1	00004335
870*		RR(J)=X	00004340
871*		CR(J)=-Y	00004345
872*		GO TO 1011	00004350
873*	501	DISC=A(2)**2-4.*A(1)	00004355
874*		IF(DISC)521,541,541	00004360
875*	521	Y=SQRT(-DISC)/2.	00004365
876*		X=-A(2)/2.	00004370
877*		GO TO 421	00004375
878*	541	J=J+1	00004380
879*		RR(J)=(-A(2)+SQRT(DISC))/2.	00004385
880*		CR(J)=0.	00004390
881*		GO TO 1021	00004395
882*	601	J=J+1	00004400
883*		RR(J)=-A(1)	00004405
884*		CR(J)=0.	00004410
885*		GO TO 2001	00004415
886*	1011	B(1)=X**2+Y**2	00004420
887*		B(2)=-2.*X	00004425
888*		B(3)=1.	00004430
889*		NB=2	00004435
890*		GO TO 1041	00004440
891*	1021	B(1)=-RR(J)	00004445
892*		B(2)=1.	00004450
893*		NB=1	00004455
894*	1041	CALL DIV(A,B,N,NB,Q)	00004460
895*		DO 1061 J1=1,N	00004465
896*	1061	A(J1)=Q(J1)	00004470
897*		IF(CR(J))1081,1071,1081	00004475
898*	1071	N=N-1	00004480
899*		GO TO 201	00004485

900*	1081	N=N-2	00004490
901*		GO TO 201	00004495
902*	2001	IF (IB) 2002, 2005, 2002	00004500
903*	2002	DO 2000 I=1, NAR	00004505
904*		RR(I)=RR(I)*(2.*(IB))	00004510
905*	2000	CR(I)=CR(I)*(2.*(IB))	00004515
906*	2005	NP1=NAR+1	00004520
907*		DO 2011 I=2, NP1	00004525
908*	2011	A(I)=0.	00004530
909*		A(1)=1.	00004535
910*		NA=0	00004540
911*		J=1	00004545
912*	2021	IF (CR(J)) 2041, 2061, 2041	00004550
913*	2041	NB=2	00004555
914*		B(3)=1.	00004560
915*		B(2)=-2.*RR(J)	00004565
916*		B(1)=RR(J)**2+CR(J)**2	00004570
917*		J=J+2	00004575
918*		GO TO 2081	00004580
919*	2061	NB=1	00004585
920*		B(2)=1.	00004590
921*		B(1)=-RR(J)	00004595
922*		J=J+1	00004600
923*	2081	CALL MTALGD(A, NA, B, NB, Q)	00004605
924*		NA=NB+NA	00004610
925*		NAPL1=NA+1	00004615
926*		DO 2091 I=1, NAPL1	00004620
927*	2091	A(I)=Q(I)	00004625
928*		IF (NA-NAR) 2021, 3001, 3001	00004630
929*	3001	DO 3011 J2=1, NPL1	00004635
930*	3011	A(J2)=A(J2)*ANPP	00004640
931*	5001	RETURN	00004645
932*		END	00004650
933*		SUBROUTINE GRAD(A, N, XZ, YZ)	00004655
934*		DIMENSION A(9999), X(3), Y(3), RP(3), CP(3), RHO(3), PHI(3)	00004660
935*		DIMENSION ABSP(3), PR(3), PC(3)	00004665
936*		PI=3.14159265	00004670
937*		MTST=1	00004675
938*	101	XZ=0.0	00004680
939*		YZ=1.0	00004685
940*		DZ=2.	00004690
941*		RHOZ=1.	00004695
942*		PHIZ=PI/2.	00004700
943*	201	CALL POLY(N, A, RZ, CZ, PRZ, PCZ, RHOZ, PHIZ)	00004705
944*	221	SU=SQRT(PRZ**2+PCZ**2)	00004710
945*		ABSPZ=SQRT(RZ**2+CZ**2)	00004715
946*		U=2.*ABSPZ*SU	00004720
947*		PSI=ATAN(U)	00004725
948*		TOP=RZ*PCZ-CZ*PRZ	00004730
949*		BOT=-(RZ*PRZ+CZ*PCZ)	00004735
950*		CALL ARCTA(BOT, TOP, THETA)	00004740
951*		COSI=COS(THETA+PHIZ)	00004745
952*		SINE=SIN(THETA+PHIZ)	00004750

953*		IF(ABSPZ)300,5001,300	00004755
954*	300	IF(SU)301,501,301	00004760
955*	301	IF(RHOZ)321,401,321	00004765
956*	321	IF(ABSPZ/(RHOZ*SU)-1.E-7)5001,5001,701	00004770
957*	351	IF(ABSPZ/(RHOZ*SU)-10.*(-MTST))801,801,401	00004775
958*	401	DZ=DZ/8.0	00004780
959*		IM=0	00004785
960*		DO 431 I=1,3	00004790
961*		DZ=2.*DZ	00004795
962*		X(I)=XZ+DZ*COSI	00004800
963*		Y(I)=YZ+DZ*SINE	00004805
964*		RHO(I)=SQRT(X(I)**2+Y(I)**2)	00004810
965*		CALL ARCTA(X(I), Y(I), PHI(I))	00004815
966*		CALL POLY(N,A,RP(I),CP(I),PR(I), PC(I),RHO(I),PHI(I))	00004820
967*		ABSP(I)=SQRT(RP(I)**2+CP(I)**2)	00004825
968*		IF(ABSPZ-ABSP(I)) 431,431,421	00004830
969*	421	ABSPZ=ABSP(I)	00004835
970*		IM=I	00004840
971*	431	CONTINUE	00004845
972*		IF(IM) 441,441,461	00004850
973*	441	DZ=DZ/8.	00004855
974*		IF(RHOZ)443,445,443	00004860
975*	443	IF(DZ/RHOZ-1.E-7)451,451,401	00004865
976*	445	IF(DZ-1.E-7)451,451,401	00004870
977*	451	IF(SU-ABSPZ) 501,501,5001	00004875
978*	461	DZ=(2.*(IM-2))*DZ	00004880
979*		XZ=X(IM)	00004885
980*		YZ=Y(IM)	00004890
981*		PHIZ=PHI(IM)	00004895
982*		PRZ=PR(IM)	00004900
983*		PCZ=PC(IM)	00004905
984*		RHOZ=RHO(IM)	00004910
985*		RZ=RP(IM)	00004915
986*		CZ=CP(IM)	00004920
987*		GO TO 221	00004925
988*	501	DZ=1.0	00004930
989*		DTHETA=PI/10.	00004935
990*	521	THETA=0.0	00004940
991*		DO 561 I=1,20	00004945
992*		THETA=THETA+DTHETA	00004950
993*		XS=XZ+DZ*COS(PHIZ+THETA)	00004955
994*		YS=YZ+DZ*SIN(PHIZ+THETA)	00004960
995*		RHOS=SQRT(XS**2+YS**2)	00004965
996*		CALL ARCTA(XS,YS,PHIS)	00004970
997*		CALL POLY(N,A,RS,CS,PRS,PCS,RHOS,PHIS)	00004975
998*		ABSP(1)=SQRT(RS**2+CS**2)	00004980
999*		IF(ABSPZ-ABSP(1))561,561,601	00004985
1000*	561	CONTINUE	00004990
1001*		DZ=DZ/2.	00004995
1002*		IF(RHOS)563,565,563	00005000
1003*	563	IF(DZ/RHOS-1.E-7)5001,5001,521	00005005
1004*	565	IF(DZ-1.E-7)5001,5001,521	00005010
1005*	601	XZ=XS	00005015

1006*		YZ=YS	00005020
1007*		PHIZ=PHIS	00005025
1008*		RHOZ=RHOS	00005030
1009*		ABSPZ=ABSP(1)	00005035
1010*		PRZ=PRS	00005040
1011*		PCZ=PCS	00005045
1012*		RZ=RS	00005050
1013*		CZ=CS	00005055
1014*		GO TO 221	00005060
1015*	701	IF(PSI-1.E-6)711,711,351	00005065
1016*	711	IF(SU-ABSPZ)501,501,351	00005070
1017*	801	RHO(1)=RHOZ+BOT/SU**2	00005075
1018*		IF(RHO(1))901,901,816	00005080
1019*	816	PHI(1)=PHIZ+TOP/(RHOZ*SU**2)	00005085
1020*	821	CALL POLY(N,A,RZ,CZ,PRZ,PCZ,RHO(1),PHI(1))	00005090
1021*		ABSP(1)=SQRT(RZ**2+CZ**2)	00005095
1022*		IF(ABSP(1)-ABSPZ)851,881,881	00005100
1023*	841	XZ=RHOZ*COS(PHIZ)	00005105
1024*		YZ=RHOZ*SIN(PHIZ)	00005110
1025*		GO TO 5001	00005115
1026*	851	RHOZ=RHO(1)	00005120
1027*		ABSPZ=ABSP(1)	00005125
1028*		PHIZ=PHI(1)	00005130
1029*		TOP=RZ*PCZ-CZ*PRZ	00005135
1030*		BOT=-(RZ*PRZ+CZ*PCZ)	00005140
1031*		SU=SQRT(PRZ**2+PCZ**2)	00005145
1032*		IF(SU)855,501,855	00005150
1033*	855	U=2.*ABSPZ*SU	00005155
1034*		PSI=ATAN(U)	00005160
1035*		IF(ABSPZ/(RHOZ*SU)-10.**(-MTST))861,861,901	00005165
1036*	861	IF(ABSPZ/(RHOZ*SU)-1.E-7)841,841,871	00005170
1037*	871	IF(PSI-1.E-6)881,881,801	00005175
1038*	881	IF(SU-ABSPZ)501,501,901	00005180
1039*	901	DZ=ABSPZ/SU	00005185
1040*		XZ=RHOZ*COS(PHIZ)	00005190
1041*		YZ=RHOZ*SIN(PHIZ)	00005195
1042*		MTST=MTST+1	00005200
1043*		GO TO 201	00005205
1044*	5001	RETURN	00005210
1045*		END	00005215
1046*		SUBROUTINE MTALGD(AARG,NA,BARG,NB,C)	00005220
1047*		DIMENSION AARG(9999),BARG(9999),C(9999),A(101),B(101)	00005225
1048*	1	NAPL1=NA+1	00005230
1049*		DO 21 J1=1,NAPL1	00005235
1050*	21	A(J1)=AARG(J1)	00005240
1051*		NBPL1=NB+1	00005245
1052*		DO 41 J1=1,NBPL1	00005250
1053*	41	B(J1)=BARG(J1)	00005255
1054*		NCPL1=NAPL1+NBPL1-1	00005260
1055*		DO 91 J1=1,NCPL1	00005265
1056*		TEMP=0.	00005270
1057*		DO 81 J2=1,J1	00005275
1058*		IF(J2-NAPL1) 61,61,81	00005280

1059*	61	N2=J1-J2+1	00005285
1060*		IF(N2-NBPL1)71,71,81	00005290
1061*	71	TEMP=TEMP+A(J2)*B(N2)	00005295
1062*	81	CONTINUE	00005300
1063*		C(J1)=TEMP	00005305
1064*	91	CONTINUE	00005310
1065*		RETURN	00005315
1066*		END	00005320
1067*		SUBROUTINE DIV(A,B,NA,NB,Q)	00005325
1068*		DIMENSION A(9999),B(9999),Q(9999)	00005330
1069*		I1=NA-NB+1	00005335
1070*		DO 61 J1=1,I1	00005340
1071*	61	Q(J1)=0.	00005345
1072*	101	KKMAX=NA-NB+1	00005350
1073*		DO 391 KK=1,KKMAX	00005355
1074*		K=KK-1	00005360
1075*	201	TEMP=0.	00005365
1076*		IF(K-1)301,211,211	00005370
1077*	211	DO 291 JJ=1,K	00005375
1078*		J=JJ-1	00005380
1079*		I1=NB-K+J	00005385
1080*		IF(I1)291,221,221	00005390
1081*	221	I2=NA-NB-J	00005395
1082*		TEMP=TEMP+B(I1+1)*Q(I2+1)	00005400
1083*	291	CONTINUE	00005405
1084*	301	I1=NA-NB-K	00005410
1085*		I2=NA-K	00005415
1086*	391	Q(I1+1)=A(I2+1)-TEMP	00005420
1087*	5001	RETURN	00005425
1088*		END	00005430
1089*	/	START ACNM=FT05F001	
1090*		9000	00000010
1091*	2	1.2	00000015
1092*	4	2.5	00000020
1093*	6	3.85	00000025
1094*	8	5.5	00000030
1095*	10	7.5	00000035
1096*	12	9.4	00000040
1097*	14	11.5	00000045
1098*	/	STOP	
1099*	/	EOJ	

APPENDIX B: COMPUTER PROGRAM PAVEVAL

The computer program PAVEVAL calculates the allowable load-carrying capacity and the required overlay thickness for rigid and flexible pavements in terms of the value of the subgrade modulus that is determined from results of vibratory nondestructive testing and in terms of the elastic moduli of the pavement layers.

The copyright of much of the information provided herein is vested in Shell Research N.V. The information is published with the expressed permission of the Shell Oil Company and Koninklijke/Shell Laboratorium, Amsterdam (KSLA). Use of the BISAR computer program in any design procedure other than the procedure contained in this document is prohibited without written permission from KSLA. It may not be included in a library of programs maintained for commercial purposes by sellers of computer services nor used in any other way which results in payment of fees to the possessor of the program solely for the usage of the program. When reference is made to the results obtained with the program, acknowledgement should be given to KSLA.

DOCUMENTATION OF THE COMPUTER PROGRAM PAVEVAL

PROGRAM IDENTIFICATION

- a. Program Title. WES Pavement Evaluation Program
- b. Program Code Name. PAVEVAL
- c. Writer. Richard A. Weiss and Ricky Austin
- d. Organization. U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS 39180
- e. Date. July 1977
- f. Source Language. Fortran IV
- g. Abstract. Program calculates the allowable load-carrying capacity and required overlay thickness of rigid and flexible pavements for single-, dual-, and dual-tandem gear configurations.

ENGINEERING DOCUMENTATION

Narrative Description. The allowable load-carrying capacity and required overlay thickness of a pavement is calculated by the combined methods of vibratory nondestructive testing and layered elastic theory. For flexible pavements, a limiting vertical strain in the subgrade and a limiting tensile strain in the AC layer are used, while for rigid pavements, a limiting tensile stress at the bottom of the PCC layer is used to relate the allowable load-carrying capacity and the required overlay thickness to the pavement structure. The computer program PAVEVAL is used to implement the layered elastic theory method of pavement evaluation. This computer program is also used to calculate the allowable load-carrying capacity and the required overlay thickness of rigid and flexible pavements for single-wheel, dual-wheel, and dual-tandem-wheel loadings.

Method of Solution. For the calculation of the allowable load-carrying capacity, the computer program PAVEVAL iterates the wheel load until the operating stress and strain in the pavement are equal to a specified limiting value. For the calculation of the required overlay thickness, PAVEVAL iterates the overlay thickness until the operating stress and strain in the pavement are less than specified limiting values.

Program Capabilities. Three subroutines are used to calculate the limiting values of the stress and the strain in the pavement. An index called EKEY is used to select the type of limiting stress or strain condition that is to be used and the depth in the pavement at which the limiting stress or strain condition is applied.

Two subgrade subroutines of the program can be used to calculate the allowable load-carrying capacity and the required overlay thickness for flexible pavements. These subroutines express the limiting strain at the top of the subgrade in terms of the load repetitions and the value of the subgrade Young's modulus. The subroutine RNN gives the limiting strain at the top of the subgrade in terms of the total number of load repetitions and does not involve values of the subgrade Young's

modulus. A more accurate representation of the limiting vertical strain at the top of the subgrade is expressed in terms of the subgrade Young's modulus value and in terms of the yearly load repetition rate. This is done in subroutine FLEX. For AC pavements using subroutine FLEX, the values of the subgrade modulus are restricted to $500 < E_s < 30000$ psi. The computer program PAVEVAL automatically introduces the limiting tensile strain at the bottom of the AC layer.

For rigid pavements, the subroutine RPAL is used to describe the limiting value of the tensile stress at the bottom of the PCC layer. The limiting tensile stress is expressed in terms of the flexural strength (R), yearly load repetition number (YRN), and the pass-to-coverage ratio for each type of landing gear.

The computer program PAVEVAL can calculate the allowable load for a rigid or a flexible pavement and the required overlay thickness for a rigid or a flexible pavement. An index EKEY2 is used to select these possibilities. By selecting the proper choices of EKEY and EKEY2, the allowable load and the required overlay thickness for rigid and flexible pavements can be calculated for a number of specified limiting stress and strain conditions.

Printed Output. The printed output consists of the allowable load-carrying capacity or the required overlay thickness for a pavement. The vertical strain at the top of the subgrade and the tensile strain at the bottom of the AC layer of flexible pavements, as well as the tensile stress at the bottom of the PCC layer, are also printed out. In addition, all components of the stress and the strain at points on the layer interfaces in the pavement directly under each of the wheel positions are printed out.

Computer Equipment. The program PAVEVAL was developed on the IBM 360/65 computer.

INPUT GUIDE FOR COMPUTER
PROGRAM PAVEVAL

The input data instructions for this WES-600 CARDIN program are shown below. The meaning of these terms is explained in the following pages.

- Line 1 (a) TEXT
- Line 2 (b) NSYS
- Line 3 (c) EKEY, EKEY2
- Line 4 (d) AA, BB, RN, ALOAD, ALIN, CAREA, DSM, SWL
- or
- (d) ES, EA, YRN, ALOAD, ALIN, CAREA, DSM, SWL, PCRATIO
- or
- (d) DSM, FAC, YRN, R, ALOAD, ALIN, CAREA, SWL
- or
- (d) AA, BB, RN, ATHICK, ATLIN, CAREA, DSM, SWL
- or
- (d) ES, EA, YRN, ATHICK, ATLIN, CAREA, DSM, SWL, PCRATIO
- or
- (d) DSM, FAC, YRN, R, ATHICK, ATLIN, CAREA, SWL
- Line 5 (e) NLAYS ISMO IRED
- Line 6 (f) E(1), NU(1), THICK(1), AK(1) or ALK(1)
- Line 7 (f) E(2), NU(2), THICK(2), AK(2) or ALK(2)
- Line 8 (g) E(NLAYS), NU(NLAYS)
- Line 9 (h) NLOAD
- Line 10 (i) LDSTRS(1), RADIUS(1), X(1), Y(1), HOSTR(1), PSI(1)
- Line 11 (i) LDSTRS(NLOAD), RADIUS(NLOAD), X(NLOAD), Y(NLOAD),
HOSTR(NLOAD), PSI(NLOAD)
- Line 12 (j) NPOS
- Line 13 (k) LAYER(1), AX(1), AY(1), DEPTH(1), ETA(1)
- Line 14 (k) LAYER(NPOS), AX(NPOS), AY(NPOS), DEPTH(NPOS), ETA(NPOS)
- Line 15 If another problem is desired, return to line 3 and repeat process.

The meaning and example of each card type are as follows:

Card type (a) TEXT = problem identification, maximum of 80 characters.

Example line: Line No. Identifying information

Card type (b) NSYS = number of problems to run

Example line: Line No. NSYS

Card type (c) EKEY = limiting strain and stress subroutine code

1 = calls subroutine RNN

2 = calls subroutine FLEX

3 = calls subroutine RFAL

EKEY2 = pavement problem code

0 = allowable load

1 = overlay over flexible pavement

2 = overlay over rigid pavement

Example line: Line No. EKEY EKEY2

Card type (d) if EKEY = 1, EKEY2 = 0

AA = -0.1616727

BB = -2.2150779

RN = total number of load repetitions

ALOAD = initial load, lb

ALIN = load increment, lb

CAREA = contact area (πr^2), in.²

DSM = dynamic stiffness modulus, for reference

SWL = 0

Example line: Line No. AA BB RN ALOAD ALIN CAREA DSM SWL

Card type (d) if EKEY = 2, EKEY2 = 0

ES = subgrade modulus, psi

EA = asphalt modulus of existing layer

YRN = yearly load repetition number

ALOAD = initial load, lb

ALIN = load increment, lb

CAREA = contact area (πr^2), in.²

DSM = dynamic stiffness modulus, for reference

SWL = 0

PCRATIO = pass-to-coverage ratio

Example line: Line No. ES EA YRN ALOAD ALIN CAREA DSM SWL PCRATIO

Card type (d) if EKEY = 3, EKEY2 = 0

DSM = dynamic stiffness modulus, for reference

FAC = pass-to-coverage ratio

YRN = yearly load repetition number

R = flexural strength, psi

ALOAD = initial load, lb

ALIN = load increment, lb

CAREA = contact area (πr^2), in.²

SWL = 0

Example line: Line No. DSM FAC YRN R ALOAD ALIN CAREA SWL

Card type (d) if EKEY = 1, EKEY2 = 1

AA = -0.1616727

BB = -2.2150779

RN = total number of load repetitions

ATHICK = initial thickness, in.

ATLIN = thickness increment, in.

CAREA = contact area (πr^2), in.²

DSM = dynamic stiffness modulus, for reference

SWL = load on one wheel, lb

Example line: Line No. AA BB RN ATHICK ATLIN CAREA DSM SWL

Card type (d) if EKEY = 2, EKEY2 = 1

ES = subgrade modulus, psi

EA = asphalt modulus of existing layer, psi

YRN = yearly load repetition number

ATHICK = initial thickness, in.

ATLIN = thickness increment, in.

CAREA = contact area (πr^2), in.²

DSM = dynamic stiffness modulus, for reference

SWL = load on one wheel, lb

PCRATIO = pass-to-coverage ratio

Example line: Line No. ES EA YRN ATHICK ATLIN CAREA DSM SWL PCRATIO

Card type (d) if EKEY = 3, EKEY2 = 2

DSM = dynamic stiffness modulus, for reference

FAC = pass-to-coverage ratio

YRN = yearly load repetition number

R = flexural strength, psi

ATHICK = initial thickness, in.

ATLIN = thickness increment, in.

CAREA = contact area (πr^2), in.²

SWL = load on one wheel, lb, for reference

Example line: Line No. DSM FAC YRN R ATHICK ATLIN CAREA SWL

Note for card type (e) and (f)

if, EKEY = 1,2

ISMO = 0

IRED = 0

all AK(i)'s = 0

if, EKEY = 3, EKEY2 = 0

ISMO = 1

IRED = 1

AK(1) = 1000

other AK(i)'s = 0

if, EKEY = 3, EKEY2 = 1

ISMO = 1

IRED = 1

AK(1) = 0

AK(2) = 1000

other AK(i)'s = 0

Card type (e) NLAYS = number of layers in pavement system

ISMO* = 0, request for rough computational procedure

1, request for smooth computational procedure

IREED = 0, AK(i) is input in card type f

1, ALK(i) is input in card type f

Example line: Line No. NLAYS ISMO IRED

* The smooth calculation procedure is more stable but less efficient than the rough procedure and is read for systems with frictionless slip between the layers or for cases when numerical instabilities are expected.

Card type (f) E(i) = modulus of layer i

NU(i) = Poisson's ratio of layer i

THICK(i)* = thickness of layer i

AK(i)** = interface compliance

or ALK(i) = reduced interface compliance

Example line: Line No. E(i) NU(i) THICK(i) AK(i) or ALK(i)

*When coding an EKEY2 = 1 or 2 problem, set THICK (1) = 1, and layer 1 is the overlay layer.

** AK(i) values are generally very small; thus, it may be more desirable to use ALK(i) where $ALK(i) = \frac{E}{1 + \nu_i} \cdot AK(i)$. For complete adhesion between layers i and i + 1, set $AK(i) = ALK(i) = 0$. For almost frictionless slip between layers, set $\frac{E_i}{1 + \nu_i} \cdot ALK(i) > 1000$.

Card type (g) E(NLAYS) = modulus of last layer

NU(NLAYS) = Poisson's ratio of last layer

Example line: Line No. E(NLAYS) NU(NLAYS)

Card type (h) NLOAD* = number of loaded areas

Example line: Line No. NLOAD

*Single wheel enter 1, dual wheel enter 2, dual-tandem enter 4

Card type (i) Load information: one card for each load

LDSTRS(i) = vertical load in units of load for loaded
area i

RADIUS(i) = radius of loaded area i

X(i)* = abscissa of center of loaded area

Y(i)* = ordinate of center of loaded area

HOSTR(i) = horizontal load in units of load for loaded
area i (normally zero)

PSI(i) = angle of HOSTR(i) with respect to positive
X-axis in degrees (normally zero)

Example line: Line No. LDSTRS(i) RADIUS(i) X(i) Y(i) HOSTR(i) PSI(i)

*X(i) and Y(i) should always be zero.

Card type (j) NPOS = number of depths that will be used for iteration
purposes

1, rigid pavement (EKEY = 3)

2, flexible pavement (EKEY = 1 or 2)

Example line: Line No. NPOS

Card type (k) LAYER (i)* = layer number for position i

AX(i) = abscissa of position (always zero)

AY(i) = ordinate of position (always zero)

DEPTH(i)* = depth from pavement surface to position

ETA(i) = angle from which position is observed with
respect to the difference of the tangential
loading (always zero)

Example line: Line No. LAYER(i) AX(i) AY(i) DEPTH(i) ETA(i)

*if, EKEY = 3, EKEY2 = 0

LAYER(1) = 1

DEPTH(1) = THICK(1)

if, EKEY = 3, EKEY2 = 2

LAYER(1) = 2

DEPTH(1) = THICK(1) + THICK(2), when THICK(1) = 1

if, EKEY = 1,2, EKEY2 = 0

LAYER(1) = 1

LAYER(2) = last layer (NLAYS)

DEPTH(1) = THICK(1)

DEPTH(2)* = distance from pavement surface to top of
subgrade

if, EKEY = 1,2, EKEY2 = 1

LAYER(1) = 2

LAYER(2) = last layer (NLAYS)

DEPTH(1) = THICK(1) + THICK(2), where THICK(1) = 1

DEPTH(2)* = distance from pavement surface to top of
subgrade

$$*DEPTH(2) = \sum_{i=1}^{(NLAYS-1)} THICK(i)$$

PROGRAM LISTING

Data are coded, then typed into a disc file, and saved for later execution. The following listing is the sample problem with control cards.

1*	PROGRAM PAVEVL	
2* C		
3* C	COMPUTATION OF STRESSES, STRAINS AND	00010000
4* C	DISPLACEMENTS IN LAYERED ELASTIC SYSTEMS	00010010
5* C	THIS PROGRAM CALCULATES THE FOLLOWING	00010020
6* C	STRESSES, STRAINS AND DISPLACEMENTS	00010030
7* C	1) RADIAL DISPLACEMENT	00010040
8* C	2) TANGENTIAL DISPLACEMENT	00010050
9* C	3) VERTICAL DISPLACEMENTS	00010060
10* C	4) RADIAL STRESS	00010070
11* C	5) TANGENTIAL STRESS	00010080
12* C	6) VERTICAL STRESS	00010090
13* C	7) RADIAL AND TANGENTIAL STRESS	00010100
14* C	8) RADIAL AND VERTICAL STRESS	00010110
15* C	9) TANGENTIAL AND VERTICAL STRESS	00010120
16* C	10) RADIAL STRAIN	00010130
17* C	11) TANGENTIAL STRAIN	00010140
18* C	12) VERTICAL STRAIN	00010150
19* C	13) RADIAL AND TANGENTIAL STRAIN	00010160
20* C	14) RADIAL AND VERTICAL STRAIN	00010170
21* C	15) TANGENTIAL AND VERTICAL STRAIN	00010180
22* C	MASTERPROGRAM	00010190
23* C		00010200
24* C	PURPOSE	00010210
25* C		00010220
26* C		00010230
27* C	THIS MASTERPROGRAM READS DATA WHICH	00010240
28* C	DETERMINE THE PHYSICAL BEHAVIOUR OF	00010250
29* C	THE SYSTEM OF LAYERS AND WHICH	00010260
30* C	DESCRIBE THE CONFIGURATION OF THE LOADS.	00010270
31* C	FOR EACH SYSTEM THE REQUIRED STRESSES	00010280
32* C	STRAINS AND DISPLACEMENTS ARE READ IN.	00010290
33* C	THEN THE COORDINATES OF EACH POSITION	00010300
34* C	ARE READ.	00010310
35* C	FOR A COMPLETE INPUT-DESCRIPTION SEE	00010320
36* C	GROUP EXTERNAL REPORT AMSR. .72	00010330
37* C	SYSTEM DATA ARE OUTPUTTED BY	00010340
38* C	1) SYSTEM	00010350
39* C	AFTER SUBSEQUENT CALLING IN OF#	00010360
40* C	2) MACONI	00010370
41* C	4) MAZCON	00010380
42* C	5) CONPNT	00010390
43* C	6) ASYMPT	00010400
44* C	7) GENDAT	00010410
45* C	8) INGRAL	00010420
46* C	THE STRESSES, STRAINS AND DISPLACEMENTS	00010430
47* C	ARE CALCULATED AND AFTER SUBSEQUENT	00010440
48* C	CALLING IN OF#	00010450
49* C	9) CALC	00010460
50* C	10) OUTPUT	00010470
51* C	11) JACOBI	00010480
	12) ESORT	00010490

```

52* C MAIN OUTPUTS OR HAS ALREADY OUTPUTTED# 00010500
53* C -ALL STRESSES,STRAINS AND DISPLACEMENTS, 00010510
54* C INDUCED BY EACH LOAD SEPARATELY AND 00010520
55* C EXPRESSED IN CYLINDRICAL COMPONENTS. 00010530
56* C -ALL TOTAL STRESSES STRAINS AND DISPLACE- 00010540
57* C MENTS EXPRESSED IN CARTESIAN COMPONENTS. 00010550
58* C -ALL PRINCIPAL TOTAL STRESSES AND STRAINS, 00010560
59* C WITH THEIR PRINCIPLE DIRECTIONS. 00010570
60* C -ALL MAXIMUM TOTAL SHEAR STRESSES AND 00010580
61* C STRAINS,WITH THEIR PRINCIPLE DIRECTIONS 00010590
62* C -THE MIDPOINTS OF THE THREE ACCOMPANYING 00010600
63* C MOHR'S CIRCLES. 00010610
64* C -THE TOTAL STRAIN ENERGY AND STRAIN 00010620
65* C ENERGY OF DISTORTION. 00010630
66* C ----- 00010640
67* C 00010650
68* C ***** Y ***** 00010660
69* C THIS PROGRAM WAS MODIFIED TO SOLVE RIGID AND FLEXIBLE PAVEMENT 00010670
70* C PROBLEMS USING THE COMBINED METHODS OF THE BISAR PROGRAM AND 00010680
71* C VIBRATORY NONDESTRUCTIVE TESTING. THIS PROGRAM SOLVES OVERLAY 00010690
72* C THICKNESS AND ALLOWABLE LOAD PROBLEMS FOR EACH PAVEMENT TYPE. 00010700
73* C 00010710
74* C 00010720
75* C PROGRAM NAME: PAVEVAL 00010730
76* C CODED BY: RICKY AUSTIN, GEOTECHNICAL LABORATORY, 00010740
77* C WATERWAYS EXPERIMENT STATION,VICKSBURG, 00010750
78* C MISSISSIPPI 00010760
79* C COMPUTER: WATERWAYS EXPERIMENT STATION, GE600 00010770
80* C LANGUAGE: FORTRAN IV 00010780
81* C DATE COMPLETED: SEPTEMBER 1978 00010790
82* C SPECIAL REQUIREMENTS: CARDIN, 00010800
83* C REMOTE BATCH PROCESSING 00010801
84* C STORAGE: DISC 00010810
85* C 00010820
86* C 00010830
87* C ***** 00010840
88* C 00010850
89* C LOGICAL STRESS,EPS,RLOW,AID(27),N,L,N2,L2,NZEP,NZEQ 00010860
90* C INTEGER REQUEST(27),IQ(3),DATE(3),ISTRSS(27),INTV(10),IVER1(7), 00010870
91* C +IVER2(10) 00010880
92* C REAL NU,K5,MU,LDSTRS(10),HOSTR(10),LOAD,INT(17),V(15),X(10),Y(10), 00010890
93* C +A(3,3),HH(3,3),W(3),C(39),B(3,3),TEXT(15),ACCUR(3),PSI(10),AK(9), 00010900
94* C +ALK(9) 00010910
95* C DOUBLE PRECISION CZ,ELLE,ELLK 00010920
96* C COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9) 00010930
97* C +,K5(10),E(10),AL(9),THICK(9),RADIUS(10) 00010940
98* C COMMON/STRDTA/STRESS(27),EPS(17),RLOW,ST,CT,L,ACC 00010950
99* C COMMON/CONST/CZ,ELLE,ELLK,ALMBDA 00010960
100* C COMMON/CNTING/F10M1,F100,F101,F11M2,F11M1,F110,F111 00010970
101* C COMMON/TAPE/NDOUT 00010980
102* C COMMON VSTR,STRL,ITER,STRL2 00010990
103* C COMMON/RADIAL/STSL,DSM,FS,SWL 00011000
104* C DIMENSION XTEMP(100),YTEMP(100),LAY(100),AXX(100) 00011010

```

```

105*      DIMENSION AYY(100),DEP(100),ETAA(100)                                00011020
106*      INTEGER EKEY,EKEY2                                                    00011030
107*      DATA NBLANK,ISTRSS,IREF1,IREF2/                                     00011040
108*      + 'UR ','UT ','UZ ','SRR','STT','SZZ','SRT','SRZ','STZ','ERR'00011050
109*      +,'ETT','EZZ','ERT','ERZ','ETZ','UX ','UY ','SXX','SXY','SXZ','SYV'00011060
110*      +,'SYZ','EXX','EXY','EXZ','EYY','EY ','LOAD','STRS'/                 00011070
111*      DATA REQUEST/UR ','UT ','UZ ','SRR','STT','SZZ','SRT','SRZ','STZ',00011080
112*      1'ERR',                                                                00011090
113*      2'ETT','EZZ','ERT','ERZ','ETZ','UX ','UY ','SXX','SXY','SXZ','SYV',00011100
114*      3'SYZ','EXX','EXY','EXZ','EYY','EY '/                                00011110
115*      DATA IVER1,IVER2/1,2,3,6,7,13,14,4,5,8,9,10,11,12,15,16,17/        00011120
116*      DATA IDENT/'LOAD'/                                                    00011130
117* C-----00011140
118* C      THESE ARE THREE ACCURACIES                                           00011150
119* C      ACCUR(1) IS USED FOR TESTING SEVERAL                               00011160
120* C      VARIABLES AGAINST EACH OTHER.                                       00011170
121* C      ACCUR(2) IS USED FOR ABSOLUTE ACCURACY                             00011180
122* C      OF THE INTEGRATION PROCEDURE                                       00011190
123* C      ACCUR(3) IS USED FOR RELATIVE ACCURACY                             00011200
124* C      OF THE INTEGRATION PROCEDURE                                       00011210
125* C      NIN ,                                                                00011220
126* C      NOUT ARE SYMBOLIC NAMES FOR INPUT AND                             00011230
127* C      OUTPUT MEDIA RESP.                                                  00011240
128* C-----00011250
129*      ACCUR(1)=1.0E-04                                                       00011260
130*      ACCUR(2)=1.0E-4                                                       00011270
131*      ACCUR(3)=1.0E-3                                                       00011280
132*      ACC=ACCUR(1)                                                          00011290
133*      NIN=5                                                                  00011300
134*      NOUT=6                                                                00011310
135*      V2=1.414214                                                           00011320
136* C      ITER = 0, INPUT NOT COMPLETE.                                       00011330
137* C      ITER = 1, INPUT COMPLETE.                                           00011340
138* C      ISKIP = 0, ITERATION NOT COMPLETE.                                 00011350
139* C      ISKIP = 1, ITERATION COMPLETE.                                     00011360
140*      WRITE(NOUT,9000)                                                       00011370
141* C-----00011380
142* C      READ TEXT AND DATE CARD                                             00011390
143* C-----00011400
144*      READ(NIN,9010)TEXT                                                     00011410
145*      WRITE(NOUT,9020)TEXT                                                  00011420
146* C-----00011430
147* C      READ NUMBER OF SYSTEMS AND SET LOUP                               00011440
148* C-----00011450
149*      CALL NFRD(NIN,NSYS,1)                                                  00011480
150*      CALL NFRD(NIN,IO,2)                                                    00011490
151*      DO 460 ISYS=1,NSYS                                                    00011500
152*      ITER=0                                                                00011510
153*      ISKIP=0                                                                00011520
154* C      SELECT LIMITING STRAIN OR STRESS SUBROUTINE CODE AND PAVEMENT
155* C      TYPE.
156*      CALL NFRD(NIN,EKEY,1)
157*      CALL NFRD(NIN,EKEY2,2)

```

158*	GO TO(501,502,503),EKEY	00011540
159*	501 CALL RNN(AA,BB,RN,ALOAD,ALIN,CAREA)	00011550
160*	GO TO 510	00011560
161*	502 CALL FLEX(ES,EA,ALOAD,ALIN,CAREA,XS,AS,BS,YRN,PRATIO)	+0011570
162*	GO TO 510	00011580
163*	503 CALL RPAL(ALOAD,ALIN,CAREA)	00011590
164*	510 CONTINUE	00011600
165*	IF(EKEY2.GT.0)GO TO 520	00011610
166*	IF(EKEY.LT.3)WRITE(NOUT,900)STRL,STRL2,ALOAD,ALIN	00011620
167*	IF(EKEY.EQ.3)WRITE(NOUT,901)FS,DSM,ALOAD,ALIN,STSL	00011630
168*	GO TO 550	00011640
169*	520 CONTINUE	00011650
170*	ATHICK=ALOAD	00011660
171*	ATLIN=ALIN	00011670
172*	GO TO(521,522),EKEY2	00011680
173*	521 WRITE(NOUT,902)	00011690
174*	WRITE(NOUT,903)DSM,SWL,ATHICK,ATLIN,STRL,STRL2	00011700
175*	GO TO 550	00011710
176*	522 WRITE(NOUT,904)	00011720
177*	WRITE(NOUT,905)DSM,SWL,FS,ATHICK,ATLIN,STSL	00011730
178*	550 CONTINUE	00011740
179*	C-----	00011750
180*	C READ NUMBER OF LAYERS AND THEIR PARAMETERS	00011760
181*	C-----	00011770
182*	CALL NFRD(NIN,NLAYS,1)	
183*	CALL NFRD(NIN,ISMO,2)	
184*	CALL NFRD(NIN,IRED,2)	
185*	IF(NLAYS.EQ.1) GO TO 10	00011790
186*	M=NLAYS-1	00011800
187*	DO 315 I=1,M	00011810
188*	CALL FFRD(NIN,E(I),1)	
189*	CALL FFRD(NIN,NU(I),2)	
190*	CALL FFRD(NIN,THICK(I),2)	
191*	315 CALL FFRD(NIN,AK(I),2)	
192*	10 CALL FFRD(NIN,E(NLAYS),1)	
193*	CALL FFRD(NIN,NU(NLAYS),2)	
194*	C-----	00011840
195*	C READ NUMBER OF LOADS AND THEIR PARAMETERS	00011850
196*	C-----	00011860
197*	CALL NFRD(NIN,NLOAD,1)	
198*	NZEP = .FALSE.	00011880
199*	NZEQ = .FALSE.	00011890
200*	DO 30 I=1,NLOAD	00011900
201*	CALL FFRD(NIN,LDSTRS(I),1)	
202*	CALL FFRD(NIN,RADIUS(I),2)	
203*	CALL FFRD(NIN,X(I),2)	
204*	CALL FFRD(NIN,Y(I),2)	
205*	CALL FFRD(NIN,HOSTR(I),2)	
206*	CALL FFRD(NIN,PSI(I),2)	
207*	IF(EKEY2.GT.0)GO TO 560	00011920
208*	LDSTRS(I)=ALOAD	00011930
209*	560 CONTINUE	00011940
210*	PSI(I)=.0174533*PSI(I)	00011950

```

211*          IF(LDSTRS(I).GT.ACCUR(1)) NZEP = .TRUE.          00011960
212*          IF(HOSTR(I).GT.ACCUR(1)) NZEQ = .TRUE.          00011970
213*          IF(IDENT.EQ.IREF1) GO TO 20                      00011980
214*          IF(IDENT.NE.IREF2) WRITE(NOUT,9040) LDSTRS(I),HOSTR(I) 00011990
215*          GO TO 30                                           00012000
216*      20      LDSTRS(I) = LDSTRS(I)/(3.14159*RADIUS(I)*RADIUS(I)) 00012010
217*          HOSTR(I) = HOSTR(I)/(3.14159*RADIUS(I)*RADIUS(I)) 00012020
218*      30      CONTINUE                                       00012030
219* C-----00012040
220* C          TEST ON OBVIOUS MISTAKES IN SYSTEM'S DATA-00012050
221* C          CARDS.                                          00012060
222* C          WHEN IRED > 0 THE REDUCED SPRINGCOMPLIAN- 00012070
223* C          CE WAS READ.                                    00012080
224* C          A NON-VANISHING SLIPRESISTANCE IS SUBSTI- 00012090
225* C          TUTED TO PREVENT RIGID-BODY MOTION OF THE 00012100
226* C          TOPLAYERS                                     00012110
227* C-----00012120
228*      DO 50 J = 1,NLAYS                                     00012130
229*          IF((1.0-NU(J)).LT.ACCUR(1)) GO TO 410             00012140
230*          IF(E(J).LT.ACCUR(1)) GO TO 420                     00012150
231*          IF(J.EQ.NLAYS) GO TO 50                            00012160
232*          IF(IRED.EQ.0) GO TO 40                              00012170
233*          ALK(J) = AK(J)                                       00012180
234*          IF(ALK(J).LT.1000.0.OR..NOT.NZEQ) GO TO 50         00012190
235*          ALK(J) = 1000.0                                       00012200
236*          AK(J) = 1000.0                                       00012210
237*          GO TO 50                                              00012220
238*      40      ALK(J) = AK(J)*E(J)/(1.0+NU(J))                 00012230
239*          IF(ALK(J).LT.1000.0.OR..NOT.NZEQ) GO TO 50         00012240
240*          ALK(J) = 1000.0                                       00012250
241*          AK(J) = ALK(J)*(1.0+NU(J))/E(J)                     00012260
242*      50      CONTINUE                                       00012270
243* C-----00012280
244* C          OUTPUT OF ALL PHYSICAL DATA OF SYSTEM          00012290
245* C          AND LOADS BY CALLING IN SYSTEM.                 00012300
246* C-----00012310
247*      CALL SYSTEM(ISYS,E,NU,THICK,AK,NLAYS,M,NLOAD,LDSTRS,HOSTR,ALK, 00012320
248*      + RADIUS,X,Y,PSI,ISMO,IRED)                             00012330
249*      IF(.NOT.NZEP.AND..NOT.NZEQ) GO TO 430                  00012340
250* C-----00012350
251* C          CALCULATION OF CONSTANTS USED IN SUBROU-        00012360
252* C          TIME MATRIX TO BUILT UP VARIOUS MATRICES        00012370
253* C          BY CALLING IN MACONI.                             00012380
254* C-----00012390
255*      CALL MACONI(ISMO,ALK,NEWSYS)                             00012400
256*      60 IF(NEWSYS.EQ.0) GO TO 70                             00012410
257*      CALL SYSTEM(ISYS,E,NU,THICK,AK,NLAY,M,NLOAD,LDSTRS,HOSTR,ALK, 00012420
258*      + RADIUS,X,Y,PSI,ISMO,IRED)                             00012430
259* C-----00012440
260* C          READ STRESSES,STRAINS AND DISPLACEMENTS        00012450
261* C          TO BE CALCULATED.                                00012460
262* C-----00012470
263*      70      CONTINUE                                       00012480

```

264*	DO 90 I=1,27	00012490
265*	IF(REQUEST(I).EQ.NBLANK) GO TO 80	00012500
266*	IF(REQUEST(I).NE.ISTRSS(I)) WRITE(NOUT,9070) ISTRSS(I)	00012510
267*	AID(I)=.TRUE.	00012520
268*	GO TO 90	00012530
269*	80 AID(I)=.FALSE.	00012540
270*	90 CONTINUE	00012550
271*	C-----	00012560
272*	C CONSYS DETERMINES FOR EACH SYSTEM WHICH	00012570
273*	C STRESSES, STRAINS AND DISPLACEMENTS WILL	00012580
274*	C BE CALCULATED.	00012590
275*	C-----	00012600
276*	CALL CONSYS(AID,NZEP,NZEQ,N,L)	00012610
277*	C-----	00012620
278*	C READ NUMBER OF POSITIONS AND SET LOOP	00012630
279*	C-----	00012640
280*	100 CALL NFRD(NIN,NPOS,1)	
281*	590 CONTINUE	00012660
282*	580 CONTINUE	00012670
283*	IF(ISKIP.EQ.0)GO TO 88	00012680
284*	NPOS=NPOS3	00012690
285*	88 CONTINUE	00012700
286*	DO 400 IPOS=1,NPOS	00012710
287*	N2 = N	00012720
288*	L2 = L	00012730
289*	DO 110 I=1,3	00012740
290*	DO 110 J=1,3	00012750
291*	110 A(I,J)=0.0	00012760
292*	C-----	00012770
293*	C READ POINT COORDINATES AND LAYERNUMBER.	00012780
294*	C-----	00012790
295*	IF(ITER.GT.0)GO TO 570	00012800
296*	CALL NFRD(NIN,LAY(IPOS),1)	
297*	CALL FFRD(NIN,AXX(IPOS),2)	
298*	CALL FFRD(NIN,AYY(IPOS),2)	
299*	CALL FFRD(NIN,DEP(IPOS),2)	
300*	CALL FFRD(NIN,ETAA(IPOS),2)	
301*	570 CONTINUE	00012820
302*	XTEMP(IPOS)=AXX(IPOS)	00012830
303*	YTEMP(IPOS)=AYY(IPOS)	00012840
304*	LAYER=LAY(IPOS)	00012850
305*	AX=AXX(IPOS)	00012860
306*	AY=AYY(IPOS)	00012870
307*	DEPTH=DEP(IPOS)	00012880
308*	ETA=ETAA(IPOS)	00012890
309*	ETA=.0174533*ETA	00012900
310*	IF(NLAYS.EQ.1) LAYER=1	00012910
311*	WRITE(NOUT,9090) IPOS,LAYER,AX,AY,DEPTH	00012920
312*	TMIN=1.0E+10	00012930
313*	IF(NLAYS.EQ.1) GO TO 130	00012940
314*	J=LAYER+1	00012950
315*	J=MINO(J,M)	00012960
316*	DO 120 I=1,J	00012970

```

317*          IF(THICK(I) LT.TMIN) TMIN=THICK(I)          00012980
318* 120      CONTINUE          00012990
319* 130      UX=0.0          00013000
320*          UY=0.0          00013010
321*          UZ=0.0          00013020
322*          MU=NU(LAYER)          00013030
323*          FT=(1.0+MU)/E(LAYER)          00013040
324* C-----          00013050
325* C          SET LOOP FOR NUMBER OF LOADS.          00013060
326* C-----          00013070
327*          DO 330 I=1,NLOAD          00013080
328*          DO 140 J=1,17          00013090
329* 140          INT(J)=0.0          00013100
330*          DO 150 J=1,27          00013110
331* 150          STRESS(J)=AID(J)          00013120
332* C          COMPUTES THE LIMITING ASPHALT STRAIN AND SUBGRADE STRAIN.          00013130
333*          IF(NLAYS.EQ.1) GO TO 160          00013140
334* C-----          00013150
335* C          CALCULATION OF CONSTANTS NEEDED FOR THE          00013160
336* C          EVALUATION OF THE CHARACTERISTIC FUNCTI-          00013170
337* C          ONS IN MATRIX BY CALLING IN MA2CON.          00013180
338* C          MA2CON.          00013190
339* C-----          00013200
340*          CALL MA2CON(TMIN,I,ISMO,ALK)          00013210
341* C-----          00013220
342* C          DETERMINATION OF POINT COORDINATES IN THE          00013230
343* C          CYLINDRICAL COORDINATE SYSTEM WITH LOAD-          00013240
344* C          AXIS AS AXIS OF SYMMETRY.          00013250
345* C-----          00013260
346* 160          IF(X(I).EQ.AX.AND.Y(I).EQ.AY) GO TO 170          00013270
347*          THETA=ATAN2((AY-Y(I)),(AX-X(I)))-PSI(I)          00013280
348*          GO TO 180          00013290
349* 170          THETA=ETA-PSI(I)          00013300
350* 180          RADDIS=SQRT((AX-X(I))**2+(AY-Y(I))**2)          00013310
351*          WRITE(NOUT,9100) I,RADDIS,THETA          00013320
352*          R=RADDIS/RADIUS(I)          00013330
353*          Z=DEPTH/RADIUS(I)          00013340
354*          IF(NLAYS.EQ.1) GO TO 230          00013350
355* 190          IF(LAYER.GT.1) GO TO 210          00013360
356*          IF(Z.GT.-ACCUR(1).AND.Z.LT.(H(1)+ACCUR(1))) GO TO 230          00013370
357* 200          WRITE(NOUT,9110)          00013380
358*          GO TO 400          00013390
359* 210          IF(LAYER LT.NLAYS) GO TO 220          00013400
360*          IF(Z.GT.(H(M)-ACCUR(1))) GO TO 230          00013410
361*          GO TO 200          00013420
362* 220          IF(Z.GT.(H(LAYER-1)-ACCUR(1)).AND.Z.LT.(H(LAYER)+ACCUR(1)))          00013430
363*          ) GO TO 230          00013440
364*          GO TO 200          00013450
365* 230          RADI=RADIUS(I)          00013460
366*          LOAD=LDSTRS(I)          00013470
367*          HSTRS=HSTR(I)          00013480
368*          RLOW=R.LT.ACCUR(1)          00013490
369*          ST=SIN(THETA)          00013500

```

370*		CT=cos(THETA)	00013510
371*	C	-----	00013520
372*	C	CONPNT DETERMINES FOR EACH POINT-LOAD	00013530
373*	C	CONFIGURATION WHICH INTEGRALS HAVE TO BE	00013540
374*	C	CALCULATED.	00013550
375*	C	-----	00013560
376*		CALL CONPNT(R,HOSTRS,LOAD,Z,N2,L2)	00013570
377*		IF(LAYER.NE.1) GO TO 250	00013580
378*		CZ = DBLE(Z)	00013590
379*		IF(Z.LT.ACCUR(1).AND.ABS(R-1.0).LT.ACCUR(1)) GO TO 240	00013600
380*	C	-----	00013610
381*	C	ASYMPT DETERMINES THE LIPSCHITZ-HANKEL	00013620
382*	C	INTEGRALS NEEDED FOR THE ASYMPTOTIC PART	00013630
383*	C	OF THE INTEGRALS, FOR POINTS IN THE TOP-	00013640
384*	C	LAYER ONLY.	00013650
385*	C	-----	00013660
386*		CALL ASYMPT(R,ACCUR(1))	00013670
387*		GO TO 250	00013680
388*	C	-----	00013690
389*	C	FOR POINTS AT THE RIM OF THE LOAD THE	00013700
390*	C	LIPSCHITZ-MANKEL INTEGRALS CAN BE GIVEN	00013710
391*	C	DIRECTLY.	00013720
392*	C	-----	00013730
393*	240	F10M1 = 0.63662	00013740
394*		F100 = 0.5	00013750
395*		F11M1 = 0.5	00013760
396*		F11M2 = 0.424413	00013770
397*		F101 = 0.0	00013780
398*		F110 = 0.0	00013790
399*		F111 = 0.0	00013800
400*	C	-----	00013810
401*	C	COMPUTATION OF THE REQUIRED INTEGRALS BY	00013820
402*	C	CALLING IN GENDAT AND INGRAL	00013830
403*	C	-----	00013840
404*	250	INTT = 0	00013850
405*		DO 260 J = 1,17	00013860
406*		INT(J) = 0.0	00013870
407*	260	CONTINUE	00013880
408*		DO 270 J = 1,10	00013890
409*		INTV(J) = 0	00013900
410*		K = IVER2(J)	00013910
411*		IF(.NOT.EPS(K)) GO TO 270	00013920
412*		INTV(J) = K	00013930
413*		INTT = INTT+1	00013940
414*	270	CONTINUE	00013950
415*		IF(INTT.EQ.0) GO TO 280	00013960
416*		IF(NLAYS.NE.1) CALL GENDAT(1,NZEROS,R,ACC)	00013970
417*		CALL INGRAL(2,INTV,INTT,INT)	00013980
418*	280	INTT = 0	00013990
419*		DO 290 J = 1,7	00014000
420*		INTV(J) = 0	00014010
421*		K = IVER1(J)	00014020
422*		IF(.NOT.FPS(K)) GO TO 290	00014030


```

423*          INTV(J) = K                                00014040
424*          INTT = INTT+1                              00014050
425* 290      CONTINUE                                  00014060
426*          IF(INTT.EQ.0) GO TO 300                    00014070
427*          IF(NLAYS.NE.1) CALL GENDAT(0,NZEROS,R,ACC) 00014080
428*          CALL INGRAL(1,INTV,INTT,INT)               00014090
429* 300      PSIO = PSI(I)                              00014100
430* C-----
431* C          CALC COMPUTES AND OUTPUTS THE STRESSES, 00014110
432* C          STRAINS AND DISPLACEMENTS, INDUCED BY EACH 00014120
433* C          LOAD SEPARATELY.                        00014130
434* C-----
435*          CALL CALC(INT,V,R,MU,RADI,FT,LOAD,HOSTRS,PSIO,Z) 00014140
436*          IF(.NOT.N2) GO TO 330                      00014150
437* C-----
438* C          COMPUTATION AND SUMMATION OF CARTESIAN 00014160
439* C          COORDINATES. THE USED COORDINATE SYSTEM IS 00014170
440* C          THE ONE WHEREIN POINTCOORDINATES WERE 00014180
441* C          STATED.                                00014190
442* C-----
443*          UZ =UZ+V(3)                                00014200
444*          IF(ABS(RADDIS).LT.ACCUR(1)) GO TO 310      00014210
445*          CT = (AX-X(I))/RADDIS                      00014220
446*          ST = (AY-Y(I))/RADDIS                      00014230
447*          GO TO 320                                  00014240
448* 310      CT =COS(ETA)                                00014250
449*          ST =SIN(ETA)                                00014260
450* 320      CT2 =CT*CT                                  00014270
451*          ST2 =ST*ST                                  00014280
452*          STCT =ST*CT                                  00014290
453*          A(1,1)=A(1,1)+V(4)*CT2+V(5)*ST2-2.0*V(7)*STCT 00014300
454*          A(1,2)=A(1,2)+V(7)*(CT2-ST2)+(V(4)-V(5))*STCT 00014310
455*          A(1,3)=A(1,3)+V(8)*CT-V(9)*ST              00014320
456*          A(2,1)=A(1,2)                              00014330
457*          A(2,2)=A(2,2)+V(4)*ST2+V(5)*CT2+2.0*V(7)*STCT 00014340
458*          A(2,3)=A(2,3)+V(8)*ST+V(9)*CT              00014350
459*          A(3,1)=A(1,3)                              00014360
460*          A(3,2)=A(2,3)                              00014370
461*          A(3,3)=A(3,3)+V(6)                         00014380
462*          UX =UX+V(1)*CT-V(2)*ST                    00014390
463*          UY =UY+V(1)*ST+V(2)*CT                    00014400
464* 330      CONTINUE                                  00014410
465*          TRACE=A(1,1)+A(2,2)+A(3,3)                 00014420
466*          AB =((1.0+MU)/E(LAYER))                   00014430
467*          AC =MU*TRACE/E(LAYER)                     00014440
468*          DO 350 I=1,3                               00014450
469*            DO 340 J=1,3                             00014460
470*              B(I,J)=AB*A(I,J)                      00014470
471*              IF(I.NE.J) GO TO 340                  00014480
472*              B(I,J)=B(I,J)-AC                      00014490
473* 340      CONTINUE                                  00014500
474* 350      CONTINUE                                  00014510
475* C-----

```

476*	C		OUTPUT FOR TOTAL STRESSES, STRAINS AND	00014570
477*	C		DISPLACEMENTS BY THREE TIMES CALLING IN	00014580
478*	C		OUTPUT.	00014590
479*	C		-----	00014600
480*		IF(ISKIP.EQ.0)GO TO 363		00014610
481*		WRITE(NOUT,9120)		00014620
482*	363	CONTINUE		00014630
483*		EPS(1)=STRESS(18)		00014640
484*		EPS(2)=STRESS(21)		00014650
485*		EPS(3)=STRESS(6)		00014660
486*		EPS(4)=STRESS(22)		00014670
487*		EPS(5)=STRESS(20)		00014680
488*		EPS(6)=STRESS(19)		00014690
489*		C(1)=A(1,1)		00014700
490*		C(2)=A(2,2)		00014710
491*		C(3)=A(3,3)		00014720
492*		C(4)=A(2,3)		00014730
493*		C(5)=A(1,3)		00014740
494*		C(6)=A(1,2)		00014750
495*		IF(ISKIP.EQ.0)GO TO 361		00014760
496*		CALL OUTPUT(EPS,C,6,1)		00014770
497*	361	CONTINUE		00014780
498*		EPS(1)=STRESS(23)		00014790
499*		EPS(2)=STRESS(26)		00014800
500*		EPS(3)=STRESS(12)		00014810
501*		EPS(4)=STRESS(27)		00014820
502*		EPS(5)=STRESS(25)		00014830
503*		EPS(6)=STRESS(24)		0 4840
504*		C(1)=B(1,1)		0 4850
505*		C(2)=B(2,2)		00014860
506*		C(3)=B(3,3)		00014870
507*		C(4)=B(2,3)		00014880
508*		C(5)=B(1,3)		00014890
509*		C(6)=B(1,2)		00014900
510*		IF(ISKIP.EQ.0)GO TO 362		00014910
511*		CALL OUTPUT(EPS,C,6,2)		00014920
512*	362	CONTINUE		00014930
513*		EPS(1)=STRESS(16)		00014940
514*		EPS(2)=STRESS(17)		00014950
515*		EPS(3)=STRESS(3)		00014960
516*		C(1)=UX		00014970
517*		C(2)=UY		00014980
518*		C(3)=UZ		00014990
519*		IF(ISKIP.EQ.0)GO TO 360		00015000
520*		CALL OUTPUT(EPS,C,3,3)		00015010
521*	360	IF(.NOT.L2) GO TO 400		00015020
522*	C	-----		00015030
523*	C	JACOBI COMPUTES PRINCIPAL VALUES AND		00015040
524*	C	DIRECTIONS OF TOTAL STRESSES AND STRAINS.		00015050
525*	C	THE PRINCIPAL VALUES ARE SORTED ACCORDING		00015060
526*	C	TO MAGNITUDE BY CALLING IN ESORT.		00015070
527*	C	-----		00015080
528*		CALL JACOBI(A,HH,3,3,1,W,IQ)		00015090

```

529*          CALL ESORT(A,HH,3,3,1,W,IQ)                                00015100
530* C-----
531* C          DETERMINATION OF MAX.SHEAR STRESSES AND 00015120
532* C          STRAINS WITH THEIR DIRECTIONS AND DETERMI-00015130
533* C          NATION OF MIDPOINTS OF THE MOHR'S CIRCLE. 00015140
534* C-----
535*          DO 370 J=1,3                                                00015150
536*              C(J)=AB*A(J,J)-AC                                       00015160
537*              C(J+5)=(HH(J,1)-HH(J,3))/V2                             00015170
538*              C(J+9)=(HH(J,1)+HH(J,3))/V2                             00015180
539*              C(J+14)=(HH(J,1)-HH(J,2))/V2                            00015190
540*              C(J+18)=(HH(J,1)+HH(J,2))/V2                            00015200
541*              C(J+23)=(HH(J,2)-HH(J,3))/V2                            00015210
542*              C(J+27)=(HH(J,2)+HH(J,3))/V2                            00015220
543*          370 CONTINUE                                                00015230
544*              C(4)=0.5*(A(1,1)-A(3,3))                                00015240
545*              C(9)=0.5*(A(1,1)+A(3,3))                                00015250
546*              C(13)=0.5*(A(1,1)-A(2,2))                               00015260
547*              C(18)=0.5*(A(1,1)+A(2,2))                               00015270
548*              C(22)=0.5*(A(2,2)-A(3,3))                               00015280
549*              C(27)=0.5*(A(2,2)+A(3,3))                               00015290
550*              C(5)=0.5*(C(1)-C(3))                                    00015300
551*              C(14)=0.5*(C(1)-C(2))                                    00015310
552*              C(23)=0.5*(C(2)-C(3))                                    00015320
553*          IF(C(13).GT.C(22))GO TO 385                                00015330
554*              DO 380 I=1,9                                              00015340
555*                  C(I+30)=C(I+12)                                       00015350
556*                  C(I+12)=C(I+21)                                       00015360
557*                  C(I+21)=C(I+30)                                       00015370
558*          380 CONTINUE                                                00015380
559*          385 CONTINUE                                                00015390
560* C-----
561* C*****
562* C-----
563* C          ITER BLOCK
564*          IF(ISKIP.GT.0)GO TO 699
565*          IF(EKEY.EQ.3)GO TO 603
566* C          FLEXIBLE
567*          IF(IPOS.EQ.1)WRITE(6,906)C(1),AX,AY,DEPTH
568*          IF(IPOS.EQ.2)WRITE(6,907)B(3,3),AX,AY,DEPTH
569*          GO TO 611
570* C          RIGID
571*          603 WRITE(6,908)A(1,1),AX,AY,DEPTH
572*          611 CONTINUE
573*          IF(EKEY.EQ.3)GO TO 620
574*          IF(EKEY2.GT.0)GO TO 630
575* C          FLEXIBLE ALLOWABLE LOAD
576*          GO TO(601,604),IPOS
577*          601 IF(ABS(C(1)).LT.STRL2)GO TO 400
578* C          ASPHALT CRITERIA CONTROLLING, SUBGRADE NOT NEEDED.
579*          IF(ITER.EQ.0) CALL NFRD(5,IDNM,1)
580*          ABSC1=C(1)
581*          IF(EKEY.EQ.1)WRITE(NOUT,915)ALOAD,PSI2,AA,BB,RN,ABSC1,STRL2

```

582*		IF(EKEY.EQ.2)WRITE(NOUT,916)ALOAD,PSI2,ES,XS,AS,BS,ABSC1,STRL2	00015630
583*		TAWGHT=(2.0*NLOAD*ALOAD)/0.95	00015640
584*		WRITE(NOUT,911)TAWGHT	00015650
585*		NPOS3=NLOAD	00015660
586*		CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)	00015670
587*		ITER=1	00015680
588*		ISKIP=1	00015690
589*		GO TO 580	00015700
590*	604	IF(ABS(B(3,3)).LT.STRL)GO TO 680	00015710
591*		VSTR=B(3,3)	00015720
592*		IF(EKEY.EQ.1)WRITE(NOUT,909)ALOAD,PSI2,AA,BB,RN,VSTR,STRL	00015730
593*		IF(EKEY.EQ.2)WRITE(NOUT,910)ALOAD,PSI2,ES,XS,AS,BS,VSTR,STRL	00015740
594*		TAWGHT=(2.0*NLOAD*ALOAD)/0.95	00015750
595*		WRITE(NOUT,911)TAWGHT	00015760
596*		NPOS3=NLOAD	00015770
597*		CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)	00015780
598*		ITER=1	00015790
599*		ISKIP=1	00015800
600*		GO TO 580	00015810
601*	620	CONTINUE	00015820
602*	C	RIGID OVERLAY	00015830
603*		IF(EKEY2.NE.2)GO TO 640	00015840
604*		IF(ABS(A(1,1)).GE.STSL)GO TO 690	00015850
605*		ABSR5=ABS(A(1,1))	00015860
606*		WRITE(NOUT,912)DSM,ABSR5,STSL,THICK(1)	00015870
607*		NPOS3=NLOAD	00015880
608*		CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)	00015890
609*		ITER=1	00015900
610*		ISKIP=1	00015910
611*		GO TO 580	00015920
612*	640	CONTINUE	00015930
613*	C	RIGID ALLOWABLE LOAD	00015940
614*		IF(ABS(A(1,1)).LT.STSL)GO TO 680	00015950
615*		RSTS=A(1,1)	00015960
616*		WRITE(NOUT,913)ALOAD,PSI2,FS,DSM,RSTS,STSL	00015970
617*		TAWGHT=(2.0*NLOAD*ALOAD)/0.95	00015980
618*		WRITE(NOUT,911)TAWGHT	00015990
619*		NPOS3=NLOAD	00016000
620*		CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)	00016010
621*		ITER=1	00016020
622*		ISKIP=1	00016030
623*		GO TO 580	00016040
624*	680	CONTINUE	00016050
625*	C	INCREMENT LOAD	00016060
626*		DO 670 I7=1,NLOAD	00016070
627*		LDSTRS(I7)=ALOAD+ALIN	00016080
628*	670	LDSTRS(I7)=LDSTRS(I7)/CAREA	00016090
629*		ALOAD=ALOAD+ALIN	00016100
630*		LOAD=LDSTRS(1)	00016110
631*		PSI2=LOAD	00016120
632*		ITER=1	00016130
633*		GO TO 580	00016140
634*	630	CONTINUE	00016150

635*	C	FLEXIBLE OVERLAY	00016160
636*		GO TO(631,632),IPOS	00016170
637*	631	CONE=C(1)	00016180
638*		GO TO 400	00016190
639*	632	IF(ABS(CONE).GE.STRL2 .OR. ABS(B(3,3)).GE.STRL)GO TO 690	00016200
640*		ABC1=ABS(CONE)	00016210
641*		ABSV12=ABS(B(3,3))	00016220
642*		WRITE(NOUT,914)DSM,ABC1,STRL2,ABSV12,STRL,THICK(1)	00016230
643*		NPOS3=NLOAD	00016240
644*		CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)	00016250
645*		ITER=1	00016260
646*		ISKIP=1	00016270
647*		GO TO 580	00016280
648*	690	CONTINUE	00016290
649*	C	INCREMENT THICKNESS	00016300
650*		THICK(1)=ATHICK+ATLIN	00016310
651*		ATHICK=ATHICK+ATLIN	00016320
652*		DO 666 IT=1,NPOS	00016330
653*	666	DEP(IT)=DEP(IT)+ATLIN	00016340
654*		ITER=1	00016350
655*		GO TO 580	00016360
656*	699	CONTINUE	00016370
657*	C		00016380
658*	C	*****	00016390
659*	C		00016400
660*	C	-----	00016410
661*	C	OUTPUT FOR PRINCIPAL STRESSES,ETC,MAXIMUM	00016420
662*	C	SHEAR STRESSES,ETC AND STRAIN ENERGIES.	00016430
663*	C	-----	00016440
664*	390	WRITE(NOUT,9130)A(1,1),C(1),HH(1,1),HH(2,1),HH(3,1),	00016450
665*	+	A(2,2),C(2),HH(1,2),HH(2,2),HH(3,2),	00016460
666*	+	A(3,3),C(3),HH(1,3),HH(2,3),HH(3,3),	00016470
667*	+	(C(1),I=4,30)	00016480
668*		BX = (A(1,1)*C(1)+A(2,2)*C(2)+A(3,3)*C(3))*0.5	00016490
669*		BY = 0.6666667*AB*(C(4)*C(4)+C(13)*C(13)+C(22)*C(22))	00016500
670*		WRITE(NOUT,9200) BX,BY	00016510
671*	400	CONTINUE	00016520
672*		GO TO 460	00016530
673*	410	WRITE(NOUT,9140) J	00016540
674*		GO TO 440	00016550
675*	420	WRITE(NOUT,9180) J	00016560
676*		GO TO 440	00016570
677*	430	WRITE (NOUT,9190)	00016580
678*	C	-----	00016590
679*	C	FOR SYSTEMS FOR WHICH IT IS CLEAR THAT	00016600
680*	C	MISTAKES OCCUR IN THE INPUTCARDS,THE	00016610
681*	C	REQUEST AND POINT INPUT CARDS ARE SKIPPED.	00016620
682*	C	PROGRAM PROCEEDS BY TAKING NEXT SYSTEM.	00016630
683*	C	-----	00016640
684*	440	READ (NIN,9150)	00016650
685*		READ(NIN,9030) NPOS	00016660
686*		DO 450 I=1,NPOS	00016670
687*	450	READ (NIN ,9150)	00016680

688*	460	CONTINUE	00016690
689*		WRITE(NOUT,9160)	00016700
690*		STOP	00016710
691*	900	FORMAT(1H1,5(2H*),	00016720
692*		1'FLEXIBLE PAVEMENT ALLOWABLE ',	00016730
693*		2'LOAD ',5(2H*),60(1H*),	00016740
694*		3'STRL(SUBGRADE LIMITING STRAIN) = ',F15.8/,	00016750
695*		4'STRL2(ASPHALT LIMITING STRAIN) = ',F15.8/,	00016760
696*		5'ALOAD = ',F15.0/,ALIN = ',F15.0/,	00016770
697*		660(1H*),60(1H*),60(1H*)	
698*	901	FORMAT(1H1,5(2H*),	00016790
699*		A58HRIGID PAVEMENT ALLOWABLE LOAD,	00016800
700*		B5(2H*),60(1H*),20X,'FS = ',F6.0//,	00016810
701*	1	20X,'DSM = ',F6.0//,	00016820
702*	2	20X,'ALOAD = ',F6.0//,	00016830
703*	3	20X,'ALIN = ',F6.0//,	00016840
704*	4	20X,'STSL = ',F15.8/,60(1H*)	00016850
705*	902	FORMAT(1H1,5(2H*),19HVERLAY OV,	00016860
706*		14HERR FLEXIBLE PAVEMENT,5(2H*))	00016870
707*	903	FORMAT(60(1H*),20X,'DSM = ',F6.0//,	00016880
708*	A	20X,'SWL = ',F12.0//,	00016890
709*	1	20X,'ATHICK = ',F6.2//,	00016900
710*	2	20X,'ATLIN = ',F6.2//,	00016910
711*		44X,'LIMITING SUBGRADE STRAIN = ',F15.8/,	00016920
712*		55X,'LIMITING ASPHALT STRAIN = ',F15.8/,60(1H*)	00016930
713*	904	FORMAT(1H1,5(2H*),23HVERLAY OVER,	00016940
714*		131H RIGID PAVEMENT,5(2H*))	00016950
715*	905	FORMAT(60(1H*),20X,'DSM = ',F6.0//,	00016960
716*	A	20X,'SWL = ',F12.0//,	00016970
717*	B	20X,'FS = ',F6.0//,	00016980
718*	1	20X,'ATHICK = ',F6.2//,	00016990
719*	2	20X,'ATLIN = ',F6.2//,	00017000
720*	3	20X,'STSL = ',F15.8/,60(1H*)	00017010
721*	906	FORMAT('MAXIMUM NORMAL STRAIN = ',F15.8,3F15.2//)	00017020
722*	907	FORMAT('VERTICAL STRAIN = ',F15.8,3F15.2//)	00017030
723*	908	FORMAT('MAXIMUM NORMAL STRESS = ',F15.8,3F15.2//)	00017040
724*	909	FORMAT(20X,'ALOAD = ',F10.0//,22X,'PSI = ',F10.2//	00017050
725*		110X,'AA = ',F12.8,' BB = ',F12.8,' RN = ',F12.0,	00017060
726*		2//16(1H*),'VSTR = ',F12.8,' STRL = ',F12.8,16(1H*)	00017070
727*	910	FORMAT(20X,'ALOAD = ',OPF10.0//,22X,'PSI = ',OPF10.2//	00017080
728*		1.10X,'ES = ',OPF12.0,3X,'XS = ',	00017090
729*		20PF8.0,7X,'AS = ',OPF12.6//,'BS = ',OPF12.6//,	00017100
730*		A16(1H*),'VSTR = ',OPF12.8,	00017110
731*		3' STRL = ',OPF12.8,16(1H*)	00017120
732*	911	FORMAT(20X,'TOTAL ALLOWABLE WEIGHT = ',F10.0//)	00017130
733*	912	FORMAT(1H1,60(1H*),60(1H*),20X,'DSM = ',F6.0//,	00017140
734*	1	20X,'ABSRS = ',F15.8//,	00017150
735*	2	20X,'STSL = ',F15.8//,	00017160
736*	3	20X,'FINAL THICKNESS = ',F8.2//,	00017170
737*	4	60(1H*),60(1H*)	00017180
738*	913	FORMAT(1H1,60(1H*),60(1H*),20X,'ALOAD = ',F10.0//,	00017190
739*	1	20X,'PSI = ',F8.0//,	00017200
740*	2	20X,'FS = ',F6.0//,	00017210

```

741*      3      20X,'DSM' = ',F6.0//, 00017220
742*      4      20X,'RSTS' = ',F15.8//, 00017230
743*      5      20X,'STSL' = ',F15.8, 00017240
744*      6      /60(1H*)//,60(1H*)///// 00017250
745* 914 FORMAT(1H1,60(1H*)//,60(1H*)//, 00017260
746*      A      20X,'ABC1' = ',F15.8//, 00017270
747*      B      20X,'STR12' = ',F15.8//, 00017280
748*      1      20X,'ABSV12' = ',F15.8//, 00017290
749*      2      20X,'STR1' = ',F15.8//, 00017300
750*      3      20X,'FINAL THICKNESS' = ',FB.2//, 00017310
751*      4      60(1H*)//,60(1H*)///// 00017320
752* 915 FORMAT(//20X,'ALOAD' = ',F10.0//,22X,'PSI' = ',F10.2, 00017330
753*      110X,'AA' = ',F12.8,' BB = ',F12.8,' RN = ',F12.0, 00017340
754*      2//16(1H*),'ABSC1' = ',F12.8,' STRL2 = ',F12.8,16(1H*)///// 00017350
755* 916 FORMAT(//20X,'ALOAD' = ',OPF10.0//,22X,'PSI' = ',OPF10.2, 00017360
756*      1,10X,'ES' = ',OPF12.0,3X,'XS' = ',OPF8.0,7X,'AS' = ',OPF12.6//, 00017370
757*      2'BS' = ',OPF12.6//,16(1H*),'ABSC1' = ',OPF12.8, 00017380
758*      3' STRL2 = ',OPF12.8,16(1H*)///// 00017390
759* 9000 FORMAT(1H1,17X,11('B'),5X,'III',5X,11('S'),6X,9('A'),5X,11('R')/ 00017400
760*      + 18X,12('B'),4X,'III',4X,12('S'),5X,11('A'),4X,12('R')/ 00017410
761*      +18X,'BB',8X,'BBB III SSS',14X,'AAA',7X,'AAA RR',8X,'RRR'/ 00017420
762*      +18X,'BB',9X,'BB III SS',15X,'AA',9X,'AA RR',9X,'RR'/ 00017430
763*      +18X,'BB',7X,'BBB III SSS',14X,'AA',9X,'AA RR',8X,'RR'/ 00017440
764*      +18X,11('B'),5X,'III',4X,11('S'),5X,13('A'),3X,12('R')/ 00017450
765*      +18X,11('B'),5X,'III',5X,11('S'),4X,13('A'),3X,11('R')/ 00017460
766*      +18X,'BB',7X,'BBB III',14X,'SSS AA',9X,'AA RR',5X,'RR'/ 00017470
767*      +18X,'BB',8X,'BB III',15X,'SS AA',9X,'AA RR',6X,'RR'/ 00017480
768*      +18X,'BB',7X,'BBB III',14X,'SSS AA',9X,'AA RR',7X,'RR'/ 00017490
769*      +18X,12('B'),4X,'III',3X,13('S'),4X,'AA',9X,'AA RR',6X,'RR'/ 00017500
770*      +18X,11('B'),5X,'III',3X,12('S'),5X,'AA',9X,'AA RR',9X,'RR'//// 00017510
771*      +75X,'THIS 'BISAR' PROGRAM HAS BEEN OBTAINED FROM'/39X,'SHELL RESO 00017520
772*      +EARCH B.V.'/89X,'FOR THE SOLE USE OF'/76X, 00017530
773*      +SHELL OIL COMPANY'/76X,'HOUSTON, TEXAS' 00017540
774*      + //76X,'ALL RIGHTS ARE RESERVED. 00017550
775*      + USE OF THIS PROGRAM'/76X,'BY UNAUTHORIZED PERSONS IS PROHIBITED') 00017560
776* 9010 FORMAT(15A4) 00017570
777* 9020 FORMAT(1H1,25(/),15X,15A4) 00017580
778* 9030 FORMAT(I2,I3,I1) 00017590
779* 9040 FORMAT(' NOTE THAT ',E12.6,' AND ',E12.6,' WILL BE CONSIDERED TO B 00017600
780*      +E LOADS IN STRESS UNITS') 00017610
781* 9050 FORMAT(4E12.6) 00017620
782* 9060 FORMAT(26A3,A2) 00017630
783* 9070 FORMAT(' NOTE THAT INCORRECT SPELLING HAS NOT STOPPED THE EVALUATI 00017640
784*      +ON OF STRESS',4X,A3) 00017650
785* 9080 FORMAT(I2,4E12.6) 00017660
786* 9090 FORMAT(1H1,///52X,'POSITION NUMBER ',I2//54X,'LAYER NUMBER ',I2// 00017670
787*      +55X,'COORDINATES'//46X,'X',11X,'Y',11X,'Z'/40X,3E12.4) 00017680
788* 9100 FORMAT(/21X,'DISTANCE TO LOAD-AXIS('I2,')',34X,'THETA'/25X,E12.4, 00017690
789*      +41X,E12.4/) 00017700
790* 9110 FORMAT(///,30X,'THIS POSITION HAS BEEN OMITTED SINCE THE LAYER NUMB 00017710
791*      +ER IS INCORRECT') 00017720
792* 9120 FORMAT(/30X,'XX',10X,'YY',10X,'ZZ',10X,'VZ',10X,'XZ',10X,'XY',10X, 00017730
793*      +'UX',10X,'UY',10X,'UZ') 00017740

```

```

794* 9130 FORMAT(/' P R I N C I P A L   V A L U E S   A N D   D I R E C T I O N 00017750
795*      + S O F   T O T A L   S T R E S S E S   A N D   S T R A I N S' /15X,'N000017760
796*      +R M A L',9X,'N O R M A L',9X,'S H E A R',10X,'S H E A R',13X,'X',14X,'Y',14X,'Z' /100017770
797*      +5X,'S T R E S S',9X,'S T R A I N',9X,'S T R E S S',9X,'S T R A I N',9X,'C O M P O N E N T',6X,00017780
798*      +'C O M P O N E N T',6X,'C O M P O N E N T'/' M A X I M U M',2-15.3,30X,3F15.3/' M I N I M A X'00017790
799*      +,2E15.3,30X,3F15.3/' M I N I M U M',2E15.3,30X,3F15.3/' M A X I M U M',30X,2E100017800
800*      +5.3,3F15.3/8X,E15.3,45X,3F15.3/' M I N I M A X',30X,2E15.3,3F15.3/8X, 00017810
801*      +E15.3,45X,3F15.3/' M I N I M U M',30X,2E15.3,3F15.3/8X,E15.3,45X,3F15.3)00017820
802* 9140 FORMAT(' THE PROBLEM CANNOT BE SOLVED,NU(' ,I2,' ) EQUALS ONE') 00017830
803* 9150 FORMAT(/) 00017840
804* 9160 FORMAT(1H1) 00017850
805* 9170 FORMAT(A4,6X,6F10.0) 00017860
806* 9180 FORMAT(' THE PROBLEM CANNOT BE SOLVED,E(' ,I2,' ) EQUALS ZERO') 00017870
807* 9190 FORMAT(' SYSTEM SKIPPED NO LOADS') 00017880
808* 9200 FORMAT(1H0,13X,' STRAIN ENERGY',E11.4/' STRAIN ENERGY OF DISTORTIO00017890
809*      +N',E11.4) 00017900
810*      END 00017910
811*      SUBROUTINE SYSTEM(ISYS,E,NU,THICK,AK,NLAYS,M,NLOAD,LDSTRS,HOSTR, 00017920
812*      IALK,RADIUS,X,Y,PSI,ISMD,IRED) 00017930
813* C-----00017940
814* C      THIS SUBROUTINE OUTPUTS ALL PHYSICAL DATA 00017950
815* C      OF THE MULTI-LAYERED SYSTEM AND ALL DATA 00017960
816* C      ON CONFIGURATION AND MAGNITUDE OF THE 00017970
817* C      LOADS. 00017980
818* C-----00017990
819*      INTEGER ROUGH(2),SMOOTH(2),ISMTH(2) 00018000
820*      REAL E(10),NU(10),THICK(9),AK(9),ALK(9),LDSTRS(10),HOSTR(10), 00018010
821*      IRADIUS(10),X(10),Y(10),PSI(10) 00018020
822*      COMMON/TAPE/NOU 00018030
823*      DATA ROUGH,SMOOTH/'ROU','GH','SMD','OTH' / 00018040
824*      WRITE(NOUT,1001) ISYS 00018050
825*      IF(IRED.EQ.0) WRITE(NOUT,1002) 00018060
826*      IF(IRED.NE.0) WRITE(NOUT,1007) 00018070
827*      IF(NLAYS.EQ.1) GO TO 40 00018080
828*      DO 30 I=1,M 00018090
829*          IF(ISMD.EQ.1) GO TO 10 00018100
830*          ISMTH(1) = ROUGH(1) 00018110
831*          ISMTH(2) = ROUGH(2) 00018120
832*          IF(ALK(I).LT.100.0) GO TO 20 00018130
833*      10  ISMTH(1) = SMOOTH(1) 00018140
834*          ISMTH(2) = SMOOTH(2) 00018150
835*      20  WRITE(NOUT,1003) I,ISMTH(1),ISMTH(2),E(I),NU(I),THICK(I),AK(I) 00018160
836*      30  CONTINUE 00018170
837*      40  WRITE(NOUT,1004) NLAYS,E(NLAYS),NU(NLAYS) 00018180
838*      WRITE(NOUT,1005) 00018190
839*      DO 50 I = 1,NLOAD 00018200
840*      50  WRITE(NOUT,1006) I,LDSTRS(I),HOSTR(I),RADIUS(I),X(I),Y(I),PSI(I) 00018210
841*      1001 FORMAT(1H1,10(/),52X,'SYSTEM NUMBER',3X,I2) 00018220
842*      1002 FORMAT(5(/),8X,'LAYER',4X,'CALCULATION',2X,'YOUNG'S',4X,'POISSON'00018230
843*      1'S',3X,'THICKNESS',3X,'INTERFACE' /8X,'NUMBER',3X,'METHOD',7X,'MODU00018240
844*      2LUS',4X,'RATIO',18X,'SPRINGCOMPL' /) 00018250
845*      1003 FORMAT(10X,I2,5X,2A3,3X,4E12.4) 00018260
846*      1004 FORMAT(10X,I2,14X,2E12.4) 00018270

```



```

847* 1005 FORMAT(///8X,'LOAD',5X,'NORMAL',7X,'SHEAR',5X,'RADIUS OF',7X,'LOAD00018280
848* 1 - POSITION',6X,'SHEAR'/8X,'NUMBER',3X,'STRESS',7X,'STRESS',4X, 00018290
849* 2'LOADED AREA',6X,'X',11X,'Y',7X,'DIRECTION'/) 00018300
850* 1006 FORMAT(10X,12,2X,6E12.4) 00018310
851* 1007 FORMAT(5(/),8X,'LAYER',4X,'CALCULATION',2X,'YOUNG'S',4X,'POISSON'00018320
852* 1'S',3X,'THICKNESS',3X,' REDUCED' /8X,'NUMBER',3X,'METHOD',7X,'MODU00018330
853* 2LUS',4X,'RATIO',18X,'SPRINGCOMPL'/) 00018340
854* RETURN 00018350
855* END 00018360
856* SUBROUTINE MACON1(ISMD,ALK,NSYS) 00018370
857* C----- 00018380
858* C THIS SUBROUTINE CALCULATES CONSTANTS USED 00018390
859* C IN SUBROUTINE MATRIX TO BUILD UP VARIOUS 00018400
860* C MATRICES. 00018410
861* C THE CONSTANTS ARE STORED IN 00018420
862* C COMMON/INDATA/. 00018430
863* C NUMERICAL STABILITY OF SOLUTIONPROCEDURE 00018440
864* C FOR THE SYSTEM IS TESTED BY CALLING IN# 00018450
865* C MA2CON, 00018460
866* C MATRIX 00018470
867* C WHEN INSTABILITY HAS TO BE EXPECTED THE 00018480
868* C SMOOTH CALCULATION PROCEDURE IS CHOSEN BY 00018490
869* C TAKING ISMD = 1 AND NSYS IS SET EQUAL 1 . 00018500
870* C----- 00018510
871* REAL K1,K2,K3(10),K4(10),K5,K6,NU,II,LOAD,ACCUR(3),ALK(9) 00018520
872* COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZERDS,H(9)00018530
873* 1,K5(10),E(10),AL(9),THICK(9),RADIUS(10) 00018540
874* COMMON/INDATA/XMAX, A1(9),B1(9),C1(9),D(9),EE(9),F(9),G(9),H1(9),00018550
875* 1II(9),K1(9),K2(9),K6(10),BE(9),BU(9),BUU(9),BMU(9),B2U(9),B2UU(9),00018560
876* 2J2(9),J1,T2(10),SS(2,10),G012(9),G021(9),G022(9),G122(9), 00018570
877* 3H012(9),H022(9),H122(9),D012(9),D022(9),C011(9),C012(9),E012(9), 00018580
878* 4F012(9),F112(9),F022(9),CC(4,2,9),DD(2,2,9),FF(2,2,9),GG(2,2,10), 00018590
879* 5HH(2,2,10),RR(4,2,10),DD2(9),G20(9),G21(9),H20(9),H021(9),GG2(10),00018600
880* 6HH2(10),Q011(9),Q111(9),Q012(9),Q112(9),Q212(9),Q022(9),Q122(9), 00018610
881* 7QF0(9),QF1(9),Z011,Z111,Z211,Z012,Z112,Z212,Z312,Z021,Z121,Z022, 00018620
882* 8Z122,Z222,K4 00018630
883* COMMON/TAPE/NOUT 00018640
884* NSYS = 0 00018650
885* IF(NLAYS.EQ.1) GO TO 10 00018660
886* GG(1,1,1) = -1.0 00018670
887* GG(2,1,1) = 1.0 00018680
888* GG(1,2,1) = 1.0-2.0*NU(1) 00018690
889* GG(2,2,1) = 2.0*NU(1) 00018700
890* HH(1,1,1) = 1.0 00018710
891* HH(1,2,1) = GG(1,2,1) 00018720
892* HH(2,1,1) = 1.0 00018730
893* HH(2,2,1) = -GG(2,2,1) 00018740
894* RR(1,1,NLAYS) = 0.0 00018750
895* RR(1,2,NLAYS) = 0.0 00018760
896* RR(2,1,NLAYS) = 0.0 00018770
897* RR(2,2,NLAYS) = 0.0 00018780
898* RR(3,1,NLAYS) = 1.0 00018790
899* RR(3,2,NLAYS) = 0.0 00018800

```

900*	RR(4,1,NLAYS) = 0.0	00018810
901*	RR(4,2,NLAYS) = 1.0	00018820
902*	SS(1,NLAYS) = 0.0	00018830
903*	SS(2,NLAYS) = 1.0	00018840
904*	GG2(1) = 1.0	00018850
905*	HH2(1) = -1.0	00018860
906*	10 K5(1)=1.0-2.0*NU(1)	00018870
907*	IF(NLAYS.EQ.1) GO TO 70	00018880
908*	K = 0	00018890
909*	K6(1) = 4.0*(1.0-NU(1))	00018900
910*	DO 30 J=1,M	00018910
911*	K1(J)=(1.0+NU(J+1))*E(J)/((1.0+NU(J))*E(J+1))	00018920
912*	K2(J)=1.0-K1(J)	00018930
913*	K3(J)=NU(J+1)-NU(J)*K1(J)	00018940
914*	K4(J)=8.0*NU(J)*NU(J+1)	00018950
915*	K5(J)=1.0-2.0*NU(J)	00018960
916*	K6(J+1) = 4.0*(1.0-NU(J+1))	00018970
917*	A1(J)= K6(J)-K2(J)	00018980
918*	B1(J)= K2(J)+K1(J)*K6(J+1)	00018990
919*	C1(J)=2.0*K2(J)	00019000
920*	D (J)= K2(J)*(1.0-4.0*NU(J))	00019010
921*	EE(J)= K2(J)*(1.0+K4(J))-6.0*K3(J)	00019020
922*	F (J)= A1(J)-B1(J)	00019030
923*	G (J)= K2(J)*(1.0-K4(J))+2.0*K3(J)	00019040
924*	H1(J)=4.0*K2(J)*(NU(J+1)-NU(J))	00019050
925*	II(J)= D(J)-H1(J)	00019060
926*	30 CONTINUE	00019070
927*	K5(M+1)=1.0-2.0*NU(M+1)	00019080
928*	IF(ISMO.EQ.1) GO TO 70	00019090
929*	DO 40 I = 1,M	00019100
930*	IF(ALK(I).LT.100.0) GO TO 50	00019110
931*	40 CONTINUE	00019120
932*	GO TO 70	00019130
933*	C-----	00019140
934*	C CALCULATION OF CONSTANTS ONLY NEEDED IN	00019150
935*	C MATRIX FOR STABILITY TEST.	00019160
936*	C-----	00019170
937*	50 TMIN = 1.0E+10	00019180
938*	NTELL = 2	00019190
939*	DUMMY = 0.0	00019200
940*	LAYER = NLAYS	00019210
941*	T2(NLAYS) = 0.0	00019220
942*	DO 60 K = 1,M	00019230
943*	IF(THICK(K).LT.TMIN) TMIN = THICK(K)	00019240
944*	DUMMY = DUMMY+THICK(K)	00019250
945*	T2(K) = 2.0*THICK(K)/RADIUS(1)	00019260
946*	H(K) = DUMMY/RADIUS(1)	00019270
947*	60 CONTINUE	00019280
948*	CALL MA2CON(TMIN,1,ISMO,ALK)	00019290
949*	TX = 6.6*RADIUS(1)/TMIN	00019300
950*	XMAX = TX+1.0	00019310
951*	C-----	00019320
952*	C TEST ON NUMERICAL STABILITY OF THE SOLU-	00019330

```

953* C          TION-PROCEDURE TO BE FOLLOWED FOR THIS 00019340
954* C          SYSTEM BY CALLING IN THE MATRIX SUBROUTI- 00019350
955* C          NE WITH NTELL = 2 . 00019360
956* C          AFTER TEST THE SMOOTH OR ROUGH CALCULATI- 00019370
957* C          ON PROCEDURE IS CHOSEN. 00019380
958* C          TEST IS ONLY NECESSARY IF NOT DIRECTLY 00019390
959* C          THE SMOOTH CALCULATION PROCEDURE HAS BEEN 00019400
960* C          CHOSEN BY ISMO=1. 00019410
961* C-----00019420
962*          CALL MATRIX(TX,1,NTELL) 00019430
963*          IF(NTELL.EQ.2) GO TO 70 00019440
964*          ISMO = 1 00019450
965*          NSYS = 1 00019460
966*          WRITE(NOUT,1001) 00019470
967*          70 RETURN 00019480
968*          1001 FORMAT(' THE MORE STABLE SMOOTH CALCULATION PROCEDURE HAS BEEN CHO00019490
969*          ISEN.')
```

```

970*          END 00019500
971*          SUBROUTINE CONSYS(AID,NZEP,NZEQ,N,L) 00019510
972* C-----00019520
973* C          THIS SUBROUTINE DETERMINES FOR EACH SYS- 00019530
974* C          TEM THE CYLINDRICAL COMPONENTS NEEDED FOR 00019540
975* C          COMPUTATION OF THE REQUIRED CARTESIAN 00019550
976* C          COMPONENTS OF STRESSES,STRAINS AND DISPLA-00019560
977* C          CEMENT. GIVEN THIS SET OF COMPONENTS A 00019570
978* C          FURTHER SELECTION IS PERFORMED ON THE 00019580
979* C          COMPONENTS THAT CAN BE COMPUTED WITH THE 00019590
980* C          INTEGRALS. 00019600
981* C          CONSYS CALLS IN SUBROUTINE LOGSET 00019610
982* C-----00019620
983*          LOGICAL AID(27),NZEP,NZEQ,EPS(5),N,L 00019630
984*          INTEGER JARG(6,14) 00019640
985*          DATA JARG/ 00019650
986*          1 4, 5, 7,18,19,21, 8, 9,20,22, 0, 0, 10,11,13,23,24,26, 00019660
987*          214,15,25,27, 0, 0, 1, 2,16,17, 0, 0, 5,10,12, 0, 0, 0, 00019670
988*          3 4,10,12, 0, 0, 0, 4, 5,10, 0, 0, 0, 4, 6,12, 0, 0, 0, 00019680
989*          4 4, 5,12, 0, 0, 0, 16,17, 0, 0, 0, 0, 18,19,21,23,24,25, 00019690
990*          523,24,26, 0, 0, 0, 20,22,25,27, 0, 0, 0, 00019700
991*          EPS(1) = AID(18).OR.AID(19).OR.AID(21) 00019710
992*          EPS(2) = AID(20).OR.AID(22) 00019720
993*          EPS(3) = AID(23).OR.AID(24).OR.AID(26) 00019730
994*          EPS(4) = AID(25).OR.AID(27) 00019740
995*          EPS(5) = AID(16).OR.AID(17) 00019750
996*          DO 10 I = 1,5 00019760
997*             IF(.NOT.EPS(I)) GO TO 10 00019770
998*             CALL LOGSET(JARG(1,I),AID) 00019780
999*          10 CONTINUE 00019790
1000*          IF(AID(10).AND..NOT.NZEQ) AID( 4)=.TRUE. 00019800
1001*          IF(AID( 6).AND..NOT.NZEP) AID(12)=.TRUE. 00019810
1002*          IF(AID( 4)) CALL LOGSET(JARG(1, 6),AID) 00019820
1003*          IF(AID(10)) AID(11)=.TRUE. 00019830
1004*          IF(AID( 5)) AID(11)=.TRUE. 00019840
1005*          IF(AID(12)) AID( 6)=.TRUE. 00019850

```

```

1006*      IF(AID( 8)) AID(14)=.TRUE.      00019870
1007*      IF(AID(14)) AID( 8)=.TRUE.      00019880
1008*      IF(AID( 5).AND.AID( 6)) CALL LOGSET(JARG(1, 7),AID) 00019890
1009*      IF(AID(11).AND.AID(12)) CALL LOGSET(JARG(1, 8),AID) 00019900
1010*      IF(.NOT.NZEQ) GO TO 20           00019910
1011*      IF(AID( 7)) AID(13)=.TRUE.      00019920
1012*      IF(AID(13)) AID( 7)=.TRUE.      00019930
1013*      IF(AID( 9)) AID(15)=.TRUE.      00019940
1014*      IF(AID(15)) AID( 9)=.TRUE.      00019950
1015*      IF(AID( 5).AND.AID(10)) CALL LOGSET(JARG(1, 9),AID) 00019960
1016*      IF(AID( 6).AND.AID(10)) CALL LOGSET(JARG(1,10),AID) 00019970
1017*      IF(AID( 1).AND.AID( 2)) CALL LOGSET(JARG(1,11),AID) 00019980
1018*      IF(AID( 4).AND.AID( 7)) CALL LOGSET(JARG(1,12),AID) 00019990
1019*      IF(AID( 7).AND.AID(10)) CALL LOGSET(JARG(1,13),AID) 00020000
1020*      IF(AID( 8).AND.AID( 9)) CALL LOGSET(JARG(1,14),AID) 00020010
1021*      GO TO 30                         00020020
1022*  20 IF(AID( 1)) CALL LOGSET(JARG(1,11),AID) 00020030
1023*      IF(AID( 4)) CALL LOGSET(JARG(1,12),AID) 00020040
1024*      IF(AID( 8)) CALL LOGSET(JARG(1,14),AID) 00020050
1025*  30 N = .FALSE.                      00020060
1026*      L = .TRUE.                      00020070
1027*      IF(AID(3).OR.AID(6).OR.AID(12).OR.AID(16).OR.AID(17)) N=.TRUE. 00020080
1028*      DO 50 I = 18,27                 00020090
1029*          IF(AID(I)) GO TO 40          00020100
1030*          L = .FALSE.                  00020110
1031*          GO TO 50                     00020120
1032*  40 N = .TRUE.                       00020130
1033*  50 CONTINUE                          00020140
1034*      RETURN                           00020150
1035*      END                               00020160
1036*      SUBROUTINE LOGSET(I,LOG)         00020170
1037* C ----- 00020180
1038* C      THIS SUBROUTINE,CALLED IN BY CONSYS AND 00020190
1039* C      CONPNT,SETS THE LOGICAL VARIABLES LOG(K) 00020200
1040* C      TRUE FOR THE K-VALUES,STORED IN THE ARGU- 00020210
1041* C      MENT I. 00020220
1042* C ----- 00020230
1043*      LOGICAL LOG(1)                    00020240
1044*      INTEGER I(1)                      00020250
1045*      DO 10 L=1,6                       00020260
1046*          IF(I(L).EQ.0)                  00020270
1047*              K=I(L)                     00020280
1048*              LOG(K)=.TRUE.              00020290
1049*      10 CONTINUE                       00020300
1050*      20 RETURN                          00020310
1051*      END                                00020320
1052*      SUBROUTINE MA2CON(TMIN,I,ISMO,ALK) 00020330
1053* C ----- 00020340
1054* C      THIS SUBROUTINE CALCULATES CONSTANTS USED 00020350
1055* C      IN SUBROUTINE MATRIX TO BUILD UP VARIOUS 00020360
1056* C      MATRICES. THESE CONSTANTS ALL DEPENDENT 00020370
1057* C      ON ALK(J) AND / OR RADIUS(I), ARE STORED 00020380
1058* C      IN COMMON/INDATA/. 00020390

```

```

1059* C-----00020400
1060*     REAL K1,K2,K4(10),K5,K6,K11,K12,NU,I1,LOAD,ACCUR(3),ALK(9) 00020410
1061*     COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)00020420
1062*     1,K5(10),E(10),AL(9),THICK(9),RADIUS(10) 00020430
1063*     COMMON/INDATA/XMAX, A1(9),B1(9),C1(9),D(9),EE(9),F(9),G(9),H1(9),00020440
1064*     1I1(9),K1(9),K2(9),K6(10),BE(9),BU(9),BUU(9),BMU(9),B2U(9),B2UU(9),00020450
1065*     2J2(9),J1,T2(10),SS(2,10),G012(9),G021(9),G022(9),G122(9), 00020460
1066*     3H012(9),H022(9),H122(9),D012(9),D022(9),C011(9),C012(9),E012(9), 00020470
1067*     4F012(9),F112(9),F022(9),CC(4,2,9),DD(2,2,9),FF(2,2,9),GG(2,2,10), 00020480
1068*     5HH(2,2,10),RR(4,2,10),DD2(9),G20(9),G21(9),H20(9),H021(9),GG2(10),00020490
1069*     6HH2(10),QG11(9),Q111(9),Q012(9),Q112(9),Q212(9),G022(9),Q122(9), 00020500
1070*     7QF0(9),QF1(9),Z011,Z111,Z211,Z012,Z112,Z212,Z312,Z021,Z121,Z022, 00020510
1071*     8Z122,Z222,K4 00020520
1072*     XMAX =6.5*RADIUS(I)/TMIN 00020530
1073*     K = 0 00020540
1074*     DO 30 J = 1,M 00020550
1075*         AL(J) = ALK(J)/(RADIUS(I)+ALK(J)) 00020560
1076*         K12 = 1.0-AL(J) 00020570
1077*         Q011(J) = K12*B1(J) 00020580
1078*         Q111(J) = 2.0*AL(J)*NU(J+1) 00020590
1079*         Q012(J) = -K12*EE(J) 00020600
1080*         Q022(J) = K12*A1(J) 00020610
1081*         Q122(J) = AL(J)*K5(J) 00020620
1082*         QF0(J) = K12*A1(J)*B1(J) 00020630
1083*         QF1(J) = 2.0*AL(J)*((1.0-NU(J))+((1.0-NU(J+1))*K1(J)) 00020640
1084*         IF(I5MO.EQ.1) GO TO 20 00020650
1085*         IF(ALK(J).GE.100.0) GO TO 20 00020660
1086*         BE(J) = -AL(J)/(1.0-AL(J)) 00020670
1087*         BU(J) = BE(J)*2.0*NU(J+1) 00020680
1088*         BUU(J) = BU(J)*K5(J) 00020690
1089*         BMU(J) = BE(J)*K5(J) 00020700
1090*         B2U(J) = BE(J)*(K5(J)-2.0*NU(J+1)) 00020710
1091*         B2UU(J) = BE(J)*(K5(J)+2.0*NU(J+1)) 00020720
1092*         GO TO 30 00020730
1093*     20 K11 = 2.0*(NU(J)-NU(J+1)) 00020740
1094*         K = K+1 00020750
1095*         J2(K) = J 00020760
1096*         GG(1,1,K+1) = K2(J) 00020770
1097*         G012(K) = K11-K2(J)*((2.0-4.0*NU(J+1)) 00020780
1098*         G021(K) = -K12*K2(J) 00020790
1099*         G022(K) = (K11-K2(J))*K12 00020800
1100*         G122(K) = -2.0*NU(J+1)*AL(J) 00020810
1101*         HH(1,1,K+1) = -3.0+4.0*NU(J)-K1(J) 00020820
1102*         H012(K) = -2.0+2.0*NU(J)+6.0*NU(J+1)-K4(J)-(2.0-4.0*NU(J+1))* 00020830
1103*         1 K1(J) 00020840
1104*         H021(K) = HH(1,1,K+1)*K12 00020850
1105*         H022(K) = -K12*(1.0-2.0*NU(J)-6.0*NU(J+1)+K4(J)-K1(J)) 00020860
1106*         H122(K) = 2.0*AL(J)*NU(J+1) 00020870
1107*         DD(1,1,K) = -K6(J) 00020880
1108*         D012(K) = -K6(J)*K5(J) 00020890
1109*         DD(2,1,K) = -K12*K6(J) 00020900
1110*         D022(K) = -NU(J)*2.0*DD(2,1,K) 00020910
1111*         C011(K) = -1.0+4.0*NU(J) 00020920

```

```

1112*      C012(K) = 4.0*NU(J)*K5(J)                                00020930
1113*      CC(2,1,K) = -2.0                                          00020940
1114*      E012(K) = -K11                                           00020950
1115*      F012(K) = K4(J)-2.0*(NU(J)+NU(J+1))                     00020960
1116*      F112(K) = 2.0*E012(K)                                    00020970
1117*      F022(K) = 1.0-4.0*NU(J+1)                                00020980
1118*      FF(2,1,K) = 2.0                                           00020990
1119*      D02(K) = 2.0*K12/K1(J)                                    00021000
1120*      G20(K) = K12*(1.0-1.0/K1(J))                              00021010
1121*      G21(K) = AL(J)*0.5/K1(J)                                  00021020
1122*      H20(K) = K12*(1.0+1.0/K1(J))                              00021030
1123* 30 CONTINUE                                                    00021040
1124*      J1 = K                                                      00021050
1125*      DUMMY=0.0                                                  00021060
1126*      T2(NLAYS) = 0.0                                           00021070
1127*      DO 40 K = 1,M                                             00021080
1128*          DUMMY=DUMMY+THICK(K)                                    00021090
1129*          T2(K)=2.0*THICK(K)/RADIUS(I)                            00021100
1130*          H(K)=DUMMY/RADIUS(I)                                    00021110
1131*          K12 = 1.0-AL(K)                                         00021120
1132*          Q112(K) = -K12*F(K)*H(K)+Q111(K)*K5(K)                00021130
1133*          Q212(K) = AL(K)*H(K)*(2.0*NU(K+1)-K5(K))              00021140
1134* 40 CONTINUE                                                    00021150
1135*      IF(LAYER.EQ.NLAYS) GO TO 50                               00021160
1136* C-----
1137* C      THESE CONSTANTS ARE USED FOR THE ASYMPTO-                00021170
1138* C      TIC EVALUATION OF THE CHARACTERISTIC                     00021180
1139* C      FUNCTIONS IN MATRIX.                                     00021190
1140* C-----
1141*      J = LAYER                                                    00021200
1142*      RK1 = 2.0*NU(J+1)*C1(J)                                     00021210
1143*      RK2 = 2.0*NU(J+1)*A1(J)                                     00021220
1144*      RK3 = 2.0*NU(J+1)*D(J)                                     00021230
1145*      Z021 = Q011(J)*C1(J)                                       00021240
1146*      RK4 = Z021*H(J)                                             00021250
1147*      K12 = 1.0-AL(J)                                             00021260
1148*      Z011 = Q011(J)*D(J)                                       00021270
1149*      Z111 = AL(J)*(RK3-G(J)-K5(J)*B1(J))-RK4                    00021280
1150*      Z211 = AL(J)*H(J)*(B1(J)-II(J)-RK1)                        00021290
1151*      Z012 = -K12*(D(J)*EE(J)+A1(J)*G(J))                        00021300
1152*      Z112 = Q122(J)*(RK2-G(J)+RK3+EE(J))+K12*H(J)*(A1(J)*    00021310
1153*      1 H1(J)+C1(J)*EE(J)-D(J)*F(J))                            00021320
1154*      Z212 = -AL(J)*H(J)*(K5(J)*(RK1+B1(J)+II(J))-RK3+EE(J)+    00021330
1155*      1 RK2+G(J))-RK4*H(J))                                       00021340
1156*      Z312 = AL(J)*H(J)*H(J)*(B1(J)-RK1-II(J))                 00021350
1157*      Z121 = AL(J)*(RK1-B1(J)+II(J))                             00021360
1158*      Z022 = K12*(A1(J)*II(J)-EE(J)*C1(J))                     00021370
1159*      Z122 = AL(J)*(RK2+EE(J)+K5(J)*(RK1+II(J)))+RK4           00021380
1160*      Z222 = Z121*H(J)                                           00021390
1161* 50 RETURN                                                        00021400
1162*      END                                                         00021410
1163*      SUBROUTINE CONPNT(R,HOSTRS,LOAD,Z,N2,L2)                   00021420
1164* C-----

```


1218*	30	I=J+5		00021990
1219*		IF(Z.LT.ACC)	I=I-3	00022000
1220*		GO TO 50		00022010
1221*	40	I=J+11		00022020
1222*		IF(ABS(R-1.0).LT.ACC)	I=I-3	00022030
1223*	50	IF(STRESS(4).OR.STRESS(10))	CALL LOGSET(KARG(1,1),EPS)	00022040
1224*		IF(STRESS(5))	CALL LOGSET(KARG(1,2),EPS)	00022050
1225*		IF(STRESS(3))	EPS(3)=.TRUE.	00022060
1226*		IF(STRESS(11))	EPS(4)=.TRUE.	00022070
1227*		IF(STRESS(12))	CALL LOGSET(KARG(1,3),EPS)	00022080
1228*		IF(STRESS(6).AND.(Z.GT.ACC))	EPS(1)=.TRUE.	00022090
1229*		IF(.NOT.STRESS(8))	GO TO 60	00022100
1230*		IF(Z.LT.ACC)	GO TO 60	00022110
1231*		IF(R.GT.ACC)	EPS(5)=.TRUE.	00022120
1232*	60	IF(I.LT.3) GO TO 180		00022130
1233*		DO 90 J = 1,12		00022140
1234*		IF .NOT.STRESS(J))	GO TO 90	00022150
1235*		IF(JJ(J,I-2))	70,90,80	00022160
1236*	70	NERR = 1		00022170
1237*		STRESS(J) = .FALSE.		00022180
1238*		L2 = .FALSE.		00022190
1239*		GO TO 90		00022200
1240*	80	CALL LOGSET(IARG(1,J),EPS)		00022210
1241*	90	CONTINUE		00022220
1242*		IF(NERR) 160,160,100		00022230
1243*	100	IF(I-10) 110,130,120		00022240
1244*	110	WRITE(NOUT,9000)		00022250
1245*		GO TO 140		00022260
1246*	120	WRITE(NOUT,9020)		00022270
1247*		STRESS(13) = .FALSE.		00022280
1248*		GO TO 140		00022290
1249*	130	WRITE(NOUT,9010)		00022300
1250*		STRESS(13) = .FALSE.		00022310
1251*		IF(STRESS(12)) GO TO 150		00022320
1252*	140	IF(STRESS(3).OR.STRESS(6)) GO TO 150		00022330
1253*		IF(STRESS(16)) GO TO 150		00022340
1254*		IF(STRESS(17)) GO TO 150		00022350
1255*		IF(STRESS(20)) GO TO 150		00022360
1256*		IF(STRESS(22)) GO TO 150		00022370
1257*		IF(STRESS(25)) GO TO 150		00022380
1258*		IF(STRESS(27)) GO TO 150		00022390
1259*		N2 = .FALSE.		00022400
1260*	150	STRESS(18) = .FALSE.		00022410
1261*		STRESS(19) = .FALSE.		00022420
1262*		STRESS(21) = .FALSE.		00022430
1263*		STRESS(23) = .FALSE.		00022440
1264*		STRESS(24) = .FALSE.		00022450
1265*		STRESS(26) = .FALSE.		00022460
1266*	160	IF(LOAD.GT.ACC) GO TO 180		00022470
1267*		DO 170 J = 1,5		00022480
1268*	170	EPS(J) = .FALSE.		00022490
1269*	180	RETURN		00022500
1270*	9000	FORMAT(' AT THIS POINT SRR,STT,ERR AND EZZ HAVE A LOGARITHMIC SING	00022510	


```

1271*          IULARITY')                                00022520
1272* 9010 FORMAT(' AT THIS POINT SRT AND ERT HAVE A LOGARITHMIC SINGULARITY' 00022530
1273* 1)                                                00022540
1274* 9020 FORMAT(' AT THIS POINT SRR,STT,SRT,ERR,EZZ AND ERT HAVE A LOGARITHM 00022550
1275* 1MIC SINGULARITY')                                00022560
1276*          END                                        00022570
1277*          SUBROUTINE GENDAT(N,NZEROS,R,ACC)          00022580
1278* C-----
1279* C          THIS SUBROUTINE GIVES THE ZEROS OF THE 00022590
1280* C          PRODUCTS J0(XR)*J1(X) AND J1(XR)*J1(X) IN 00022600
1281* C          THE RIGHT ORDER. THE SUBSEQUENT ZEROS ARE 00022610
1282* C          STORED IN ZEROS FOR USING THEM IN INGRAL. 00022620
1283* C          THE ZEROS OF J0 AND J1 ARE STORED AS 00022630
1284* C          BZEROS IN THE BLOCK DATA. 00022640
1285* C-----
1286*          COMMON/GEDATA/BZEROS(149,2),ZEROS(298) 00022650
1287*          IF(R.LT.ACC.OR.ABS(R-1.0).LT.ACC) GO TO 40 00022660
1288*          I=1 00022670
1289*          J=1 00022680
1290*          DO 20 K=1,298 00022690
1291*              IF(I.GT.149) GO TO 30 00022700
1292*              IF(J.GT.149) GO TO 30 00022710
1293*              IF(BZEROS(I,2).LT.BZEROS(J,N+1)/R) GO TO 10 00022720
1294*              ZEROS(K) = BZEROS(J,N+1)/R 00022730
1295*              J=J+1 00022740
1296*              GO TO 20 00022750
1297*          10 ZEROS(K)=BZEROS(I,2) 00022760
1298*              I=I+1 00022770
1299*          20 CONTINUE 00022780
1300*          30 NZEROS = K-1 00022790
1301*          RETURN 00022800
1302*          40 IF(R.GT.ACC) GO TO 70 00022810
1303*          50 DO 60 I=1,149 00022820
1304*              ZEROS(I)=BZEROS(I,2) 00022830
1305*          60 CONTINUE 00022840
1306*          NZEROS=149 00022850
1307*          RETURN 00022860
1308*          70 IF(N.EQ.1) GO TO 50 00022870
1309*          DO 80 K=1,149 00022880
1310*              ZEROS(2*K-1)=BZEROS(K,1) 00022890
1311*              ZEROS(2*K) =BZEROS(K,2) 00022900
1312*          80 CONTINUE 00022910
1313*          NZEROS=298 00022920
1314*          RETURN 00022930
1315*          END 00022940
1316*          SUBROUTINE ASYMPT(R,ACC) 00022950
1317* C-----
1318* C          THIS SUBROUTINE ORGANIZES THE COMPUTATION 00022960
1319* C          OF THE ASYMPTOTIC PART OF THE INTEGRALS 00022970
1320* C          AS USED FOR THE TOP-LAYER ONLY. 00022980
1321* C          ASYMPT CALLS IN SUBROUTINE ASS 00022990
1322* C          ASYMPT CALLS IN FUNCTIONS FLLE 00030000
1323* C          FLK 00030010

```

```

1324* C                                     FLMBDA          00023050
1325* C-----
1326*      DOUBLE PRECISION DR,KACC2,C,ELLE,ELLK,FLLK          00023070
1327*      COMMON/CONST/C,ELLE,ELLK,ALMBDA          00023080
1328*      IF(R.LT.ACC) GO TO 10          00023090
1329*      DR=DBLE(R)          00023100
1330*      KACC2=((1.0D0-DR)*(1.0D0-DR)+C*C)/((1.0D0+DR)*(1.0D0+DR)+C*C)          00023110
1331*      ELLK=FLLK(KACC2)          00023120
1332*      ELLE=FLLK(KACC2)          00023130
1333*      ALMBDA = FLMBDA(DR,C,ELLK,ELLE,KACC2)          00023140
1334*      10 CALL ASS(ACC,R)          00023150
1335*      RETURN          00023160
1336*      END          00023170
1337*      DOUBLE PRECISION FUNCTION FLLK(KACC2)          00023180
1338* C-----
1339* C      THIS FUNCTIONSUBROUTINE EVALUATES THE          00023200
1340* C      COMPLETE ELLIPTIC INTEGRAL OF THE FIRST          00023210
1341* C      KIND FROM A SERIES-EXPANSION ACCORDING TO          00023220
1342* C      BYRD AND FRIEDMAN,HANDBOOK OF ELLIPTIC          00023230
1343* C      INTEGRALS FOR ENGINEERS AND PHYSICISTS,          00023240
1344* C      FORMULA 900.00 FOR KACC2.GE.0.5          00023250
1345* C      FORMULA 900.06 FOR KACC2.LT.0.5          00023260
1346* C-----
1347*      DOUBLE PRECISION KACC2,KA,M1,KACC          00023280
1348*      KA = 1.0D0-KACC2          00023290
1349*      IF(KA.GT.0.5D0) GO TO 10          00023300
1350*      FLLK=1.0D0+KA*(0.25D0+KA*(0.140625D0+KA*(0.09765625D0+KA*(0.07476800023310
1351*      10664D0+KA*(0.0605621338D0+KA*(0.050889015D0+KA*(0.0438787937D0+KA00023320
1352*      2*(0.0385653465D0+KA*(0.0343993364D0+KA*(0.0310454012D0+KA*(0.0282800023330
1353*      372353D0+KA*(0.0259790743D0+KA*(0.0240191152D0+KA*(0.0223341012D0+K00023340
1354*      4A*(0.0208699768D0)))))))))          00023350
1355*      GO TO 30          00023360
1356*      10 KACC=DSQRT(KACC2)          00023370
1357*      IF(KACC.LT.1.0D-04) GO TO 20          00023380
1358*      M1=-DLOG(KACC)          00023390
1359*      FLLK=M1*(1.0D0+KACC2*(0.25D0+KACC2*(0.140625D0+KACC2*(0.09765625D000023400
1360*      1+KACC2*(0.0747680664D0+KACC2*(0.0605621338D0+KACC2*(0.050889015D00023410
1361*      20+KACC2*(0.0438787937D0+KACC2*(0.0385653465D0+KACC2*(0.03439933600023420
1362*      34D0+KACC2*(0.0310454012D0+KACC2*(0.0282872353D0+KACC2*(0.02597900023430
1363*      40743D0+KACC2*(0.0240191152D0+KACC2*(0.0223341012D0)))))))))00023440
1364*      5+1.38629436D0+KACC2*(0.0965735903D0+KACC2*(0.0308851445D0+KACC2*(000023450
1365*      6.0149376004D0+KACC2*(0.0087663122D0+KACC2*(0.0057548877D0+KACC2*(000023460
1366*      7.0040646585D0+KACC2*(0.0030225465D0+KACC2*(0.0023351572D0+KACC2*(000023470
1367*      8.0018580703D0+KACC2*(0.0015135116D0+KACC2*(0.0012565911D0+KACC2*(000023480
1368*      9.0010599297D0+KACC2*(0.0009060596D0+KACC2*(0.0007834118D0))))))00023490
1369*      T)))          00023500
1370*      FLLK = 2.0D0*FLLK/3.1415926535D0          00023510
1371*      GO TO 30          00023520
1372*      20 FLLK = 0.0D0          00023530
1373*      30 RETURN          00023540
1374*      END          00023550
1375*      DOUBLE PRECISION FUNCTION FLE(KACC2)          00023560
1376* C-----

```

```

1377* C THIS FUNCTIONSUBROUTINE EVALUATES THE 00023580
1378* C COMPLETE ELLIPTIC INTEGRAL OF THE SECOND 00023590
1379* C KIND FROM A SERIES-EXPANSION ACCORDING TO 00023600
1380* C BYRD AND FRIEDMAN,HANDBOOK OF ELLIPTIC 00023610
1381* C INTEGRALS FOR ENGINEERS AND PHYSICISTS, 00023620
1382* C FORMULA 900.07 FOR KACC2.GE.0.5 00023630
1383* C FORMULA 900.10 FOR KACC2.LT.0.5 00023640
1384* C -----00023650
1385* DOUBLE PRECISION KACC2,KACC,KA 00023660
1386* KA = 1.000-KACC2 00023670
1387* IF(KA.GT.0.500) GO TO 10 00023680
1388* FLLE=1.000-KA*(0.2500+KA*(0.04687500+KA*(0.0195312500+KA*(0.01068100023690
1389* 1152300+KA*(0.006729126000+KA*(0.004626274100+KA*(0.003375291800+KA00023700
1390* 2*(0.002571023100+KA*(0.002023490400+KA*(0.001633968500+KA*0.00134700023710
1391* 3011200))))))))) 00023720
1392* GO TO 30 00023730
1393* 10 KACC=DSQRT(KACC2) 00023740
1394* IF(KACC.LT.1.00-04) GO TO 20 00023750
1395* FLLE=1.000-0.5000*KACC2*DLOG(KACC)*(1.000+KACC2*(0.37500+KACC2*(0 00023760
1396* 1.23437500+KACC2*(0.170898437500+KACC2*(0.134582519600+KACC2*(0.11100023770
1397* 2030578600+KACC2*(0.094508171100+KACC2*(0.082272738200+KACC2*(0.07200023780
1398* 3845653600+KACC2*(0.065358739300+KACC2*(0.059268493100+KACC2*(0.05400023790
1399* 4217201100+KACC2*(0.049959760600+KACC2*(0.046322580200+KACC2*0.043100023800
1400* 579262300))))))))) +0.2500*KACC2*(1.77258872200+KACC2*(0.22722000023810
1401* 6770700+KACC2*(0.087325481700+KACC2*(0.046178085600+KACC2*(0.02856800023820
1402* 7001200+KACC2*(0.019418973300+KACC2*(0.014058751800+KACC2*(0.01064800023830
1403* 8943400+KACC2*(0.008345589500+KACC2*(0.006716673700+KACC2*(0.00552200023840
1404* 9288800+KACC2*(0.004620493600+KACC2*(0.003922930400+KACC2*(0.00337200023850
1405* 1254900+KACC2*0.002929929800))))))))) 00023860
1406* FLLE = 2.000*FLLE/3.141592653500 00023870
1407* GO TO 30 00023880
1408* 20 FLLE = 2.000/3.141592653500 00023890
1409* 30 RETURN 00023900
1410* END 00023910
1411* FUNCTION FLMBDA(DR,C,ELLK,ELLE,KACC2) 00023920
1412* C -----00023930
1413* C THIS FUNCTIONSUBROUTINE EVALUATES THE 00023940
1414* C HEUMAN'S-LAMBD^ FUNCTION FROM A SERIES- 00023950
1415* C EXPANSION ACCORDING TO 00023960
1416* C BYRD AND FRIEDMAN,HANDBOOK OF ELLIPTIC 00023970
1417* C INTEGRALS FOR ENGINEERS AND PHYSICISTS, 00023980
1418* C FORMULA 904.00 00023990
1419* C USE IS MADE OF THE COMPLETE ELLIPTIC INTE-00024000
1420* C GRALS OF THE FIRST AND SECOND KIND ELLK 00024010
1421* C AND ELLE EVALUATED BY FLLK AND FLLE. 00024020
1422* C -----00024030
1423* DOUBLE PRECISION DR,DASIN,SUM,PHI,DS,DC,A,T,AI,KACC2,TWAI,DAR,ELLK00024040
1424* I,ELLE,E,K,C 00024050
1425* DAR = DABS(1.000-DR) 00024060
1426* IF(C.LT.DAR) GO TO 10 00024070
1427* DASIN = DAR/C 00024080
1428* PHI = 1.570796326800-DATAN(DASIN) 00024090
1429* GO TO 30 00024100

```

1430*	10 DASIN = C/DAR	00024110
1431*	IF(C.LT.(0.1D-05*DAR)) GO TO 20	00024120
1432*	PHI = DATAN(DASIN)	00024130
1433*	GO TO 30	00024140
1434*	20 PHI = DASIN	00024150
1435*	30 IF(DABS(PHI-1.5707963268D0).GT.1.0D-6) GO TO 40	00024160
1436*	FLMBDA=1.0	00024170
1437*	GO TO 60	00024180
1438*	40 DS=DSIN(PHI)	00024190
1439*	DC=DCOS(PHI)	00024200
1440*	E = ELLE	00024210
1441*	K = ELLK	00024220
1442*	FLMBDA=PHI*E	00024230
1443*	T=0.5D0*(PHI-DS*DC)	00024240
1444*	A=0.5D0*KACC2	00024250
1445*	SUM=ATN(2.0D0*K-E)	00024260
1446*	IF(SUM.LT.1.0D-07) GO TO 60	00024270
1447*	I=1	00024280
1448*	50 FLMBDA = FLMBDA-SNGL(SUM)	00024290
1449*	I=I+1	00024300
1450*	AI=I	00024310
1451*	TWAI=2.0D0*AI-1.0D0	00024320
1452*	T=0.5D0*T*TWAI/AI-0.5D0*DC*(DS**TWAI)/AI	00024330
1453*	A=0.5D0*A*(TWAI-2.0D0)*KACC2/AI	00024340
1454*	SUM=ATN(2.0D0*AI*K-TWAI*E)	00024350
1455*	IF(SUM.GT.1.0D-07) GO TO 50	00024360
1456*	60 RETURN	00024370
1457*	END	00024380
1458*	SUBROUTINE ASS(ACC,R)	00024390
1459*	C-----	00024400
1460*	C THIS SUBROUTINE COMPUTES THE LIPSCHITZ-	00024410
1461*	C HANKEL INTEGRALS I(I,J,K) FROM EXPRESSI-	00024420
1462*	C ONS IN EARLIER EVALUATED ELLIPTIC FUNCTI-	00024430
1463*	C ONS OF THE FIRST AND SECOND KIND,ELLK AND	00024440
1464*	C ELLE,AND HEUMAN'S-LAMBDA FUNCTION,ALMBDA.	00024450
1465*	C REFERENCE	00024460
1466*	C EASON,NOBLE AND SNEDDON,CERTAIN INTEGRALS	00024470
1467*	C OF LIPSCHITZ-HANKEL TYPE INVOLVING PRO-	00024480
1468*	C DUCTS OF BESSEL FUNCTIONS,PHILOSOPHICAL	00024490
1469*	C TRANSACTIONS,VOL 247,SERIES A935,APRIL	00024500
1470*	C 1955,PP 529-546.	00024510
1471*	C F10M1=I(1,0;-1)	00024520
1472*	C F100 =I(1,0;0)	00024530
1473*	C F101 =I(1,0;1)	00024540
1474*	C F11M2=I(1,1;-2)	00024550
1475*	C F11M1=I(1,1;-1)	00024560
1476*	C F110 =I(1,1;0)	00024570
1477*	C F111 =I(1,1;1)	00024580
1478*	C-----	00024590
1479*	C COMMON/CONST/C,ELLE,ELLK,ALMBDA	00024600
1480*	C COMMON/CNTING/F10M1,F100,F101,F11M2,F11M1,F110,F111	00024610
1481*	C DOUBLE PRECISION DR,C,DEPR,DEMR,DC2,DRT2,DRT,DAD,DR2,DRC2,DRRT,	00024620
1482*	C IDEMR,ELLE,ELLK	00024630

1483*	EC = SNGL(C)	00024640
1484*	IF(R.LT.ACC) GO TO 20	00024650
1485*	EMR = 1.0-R	00024660
1486*	EPR = 1.0+R	00024670
1487*	C2 = EC*EC	00024680
1488*	RT2 = C2+EPR*EPR	00024690
1489*	RT = SQRT(RT2)	00024700
1490*	R2 = R*R	00024710
1491*	EMRR = 1.0-R2	00024720
1492*	DR = DBLE(R)	00024730
1493*	DEPR = 1.000+DR	00024740
1494*	DEMR = 1.000-DR	00024750
1495*	DC2 = C*C	00024760
1496*	DRT2 = DC2+DEPR*DEPR	00024770
1497*	DRT = DSQRT(DRT2)	00024780
1498*	DAD = DC2+DEMR*DEMR	00024790
1499*	DR2 = DR*DR	00024800
1500*	DRC2 = DR2+DC2	00024810
1501*	DRRT = DR*DRT	00024820
1502*	DEMRR = 1.000-DR2	00024830
1503*	F101 = 0.500*(ELLE*(1.000-DRC2)/(DAD*DRT)+ELLK/DRT)	00024840
1504*	F110 = DRT*(ELLK*(1.000+DRC2)/DRT2-ELLE)/(2.000*DR)	00024850
1505*	F111 = C*(ELLE*(1.000+DRC2)/DAD-ELLK)/(2.000*DRRT)	00024860
1506*	F10M1 = 0.500*ELLE*DRT	00024870
1507*	F100 = -0.500*C*ELLK/DRT	00024880
1508*	F11M2 = -C*(C*ELLE*DRT/(4.000*DR)-C*ELLK*(1.000+DR2+0.500*DC2)/	00024890
1509*	(2.000*DRRT))/3.000+DR*(DRT*ELLE/2.000+DEMRR*ELLK/(2.000*DRT))/	00024900
1510*	23.000+(ELLE*DRT/(2.000*DR)-DEMRR*ELLK/(2.000*DRRT))/3.000	00024910
1511*	F11M1 = 0.500*(0.500*ELLE*C*DRT/DR-ELLK*C*(1.000+DR2+0.500*DC2)/	00024920
1512*	DRRT)	00024930
1513*	HLP = R	00024940
1514*	IF(R.GT.1.0) HLP=1.0/R	00024950
1515*	IF(ABS(EMR).LT.ACC) GO TO 10	00024960
1516*	F10M1 = F10M1+0.5*(SNGL(ELLK)*EMRR/RT+SIGN(EC*ALMBDA,EMR))	00024970
1517*	F100 = F100+0.5*SIGN(ALMBDA,-EMR)	00024980
1518*	F11M2 = F11M2-EC*ALMBDA*SIGN(1.0,EMR)*EMRR/(4.0*R)	00024990
1519*	F11M2 = F11M2-EC*(HLP/2.0+HLP)/3.0	00025000
1520*	F11M1 = F11M1+SIGN(0.25,EMR)*EMRR*ALMBDA/R+0.5*HLP	00025010
1521*	IF(R.GT.1.0) GO TO 30	00025020
1522*	F10M1 = F10M1-EC	00025030
1523*	F100 = F100+1.0	00025040
1524*	GO TO 30	00025050
1525*	10 F10M1 = F10M1-0.5*EC	00025060
1526*	F100 = F100+0.5	00025070
1527*	F11M2 = F11M2-0.5*EC	00025080
1528*	F11M1 = F11M1+0.5*HLP	00025090
1529*	GO TO 30	00025100
1530*	20 AD = 1.0+EC*EC	00025110
1531*	RT = SQRT(AD)	00025120
1532*	ADRT = AD*RT	00025130
1533*	F101 = 1.0/ADRT	00025140
1534*	F110 = 0.5/ADRT	00025150
1535*	F111 = 1.5*EC/(AD*ADRT)	00025160

```

1536*      F10M1 = RT-EC                                00025170
1537*      F100 = 1.0-EC/RT                            00025180
1538*      F11M2 = 0.5*(RT-EC)                        00025190
1539*      F11M1 = 0.5*(1.0-EC/RT)                    00025200
1540*      30 RETURN                                    00025210
1541*      END                                          00025220
1542*      SUBROUTINE INGRAL(IL,INTV,INTT,INT)          00025230
1543* C-----
1544* C      THIS SUBROUTINE CONTROLS THE SIMULTANEOUS 00025250
1545* C      COMPUTATION OF A GROUP OUT OF THE 17 INTE-00025260
1546* C      GRALS.                                     00025270
1547* C      IL=1,THE GROUP WITH J0(XR)J1(X) IN INTEGR.00025280
1548* C      IL=2,THE GROUP WITH J1(XR)J1(X) IN INTEGR.00025290
1549* C      THE ELEMENTS OF INTV ARE THE NUMBERS OF 00025300
1550* C      REQUIRED INTEGRALS OF THE GROUP,           00025310
1551* C      INTT IS THE TOTAL NUMBER OF REQUIRED INTE-00025320
1552* C      GRALS IN THE GROUP.                       00025330
1553* C      THE SET OF COMPUTED INTEGRALS IS DELIVE- 00025340
1554* C      RED IN INT.                                00025350
1555* C      ACTUAL INTEGRATION BY MEANS OF A GAUSS- 00025360
1556* C      QUADRATURE IS PERFORMED BY CALLING QUAD. 00025370
1557* C      INTEGRATION PROCEEDS BY QUADRATURE OVER 00025380
1558* C      INTERVALS FROM ONE BESSEL ZERO TO THE 00025390
1559* C      NEXT. FROM THE ORIGIN TO THE FIRST BESSEL-00025400
1560* C      ZERO A LEGENDRE-GAUSS QUADRATURE OF ORDER 00025410
1561* C      8,OBTAINING DESIRED ACCURACY BY SUBSE- 00025420
1562* C      QUENT SUBDIVISION OF THE INTERVAL.        00025430
1563* C      FROM THE FIRST BESSELZERO ON A JACOBI- 00025440
1564* C      GAUSS QUADRATURE,OBTAINING DESIRED ACCU- 00025450
1565* C      RACY BY SUBSEQUENT RAISING THE ORDER      00025460
1566* C      STARTING WITH THE 4TH ORDER.              00025470
1567* C      INTEGRATION STOPS AS SOON AS TWO SUBSE- 00025480
1568* C      QUENT INTERVALS DO NOT CONTRIBUTE        00025490
1569* C      SIGNIFICANTLY.                            00025500
1570* C      INTEGRATION STOPS PREMATURELY IF#        00025510
1571* C      -IN THE FIRST INTERVAL MORE THAN 30 SUB- 00025520
1572* C      DIVISIONS ARE NEEDED.                     00025530
1573* C      -IN THE FOLLOWING INTERVALS EVEN THE 15TH 00025540
1574* C      ORDER IS NOT ACCURATE ENOUGH.             00025550
1575* C      -EVEN THE 149-TH(298 TH)INTERVAL DOES    00025560
1576* C      GIVE A NON-NEGIGIBLE CONTRIBUTION.       00025570
1577* C-----
1578* C      INTEGER ALFA,ORDER,INTV(10),INTV2(10),INTV3(10),KK(10),BETA 00025590
1579* C      REAL MIDPNT,LOWER,LOAD,NU,ACCUR(3),K5,COMP(10),FIRST(10), 00025600
1580* C      1SECOND(10),INT(17),RES(10)                00025610
1581* C      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)00025620
1582* C      1,K5(10),E(10),AL(9),THICK(9),RADIUS(10) 00025630
1583* C      COMMON/GEDATA/BZEROS(149,2),ZEROS(298)    00025640
1584* C      COMMON/CNTING/F10M1,F100,F101,F11M2,F11M1,F110,F111 00025650
1585* C      COMMON/TAPE/NOUT                            00025660
1586* C      NTELL = 0                                    00025670
1587* C      NINT = 7                                      00025680
1588* C      IF(IL.EQ.2) NINT = 10                      00025690

```

1589*	DO 1000 I = 1,NINT	00025700
1590*	KK(I) = 0	00025710
1591*	1000 CONTINUE	00025720
1592*	IF(LAYER.NE.1) GO TO 2000	00025730
1593*	C-----	00025740
1594*	C	00025750
1595*	C	00025760
1596*	C-----	00025770
1597*	DO 1190 I = 1,NINT	00025780
1598*	K = INTV(I)	00025790
1599*	IF(K.EQ.0) GO TO 1190	00025800
1600*	GO TO (1010,1020,1030,1040,1050,1060,1070,1080,1090,1100,1110,	00025810
1601*	1120,1130,1140,1150,1160,1170),K	00025820
1602*	1010 INT(K) = F100+Z*F101	00025830
1603*	GO TO 1190	00025840
1604*	1020 INT(K) = F100	00025850
1605*	GO TO 1190	00025860
1606*	1030 INT(K) = -2.0*(1.0-NU(1))*F10M1-Z*F100	00025870
1607*	GO TO 1190	00025880
1608*	1040 INT(K) = (1.0-2.0*NU(1))*F11M1-Z*F110	00025890
1609*	GO TO 1180	00025900
1610*	1050 INT(K) = Z*F111	00025910
1611*	GO TO 1190	00025920
1612*	1060 INT(K) = F100-Z*F101	00025930
1613*	GO TO 1190	00025940
1614*	1070 INT(K) = -2.0*(1.0-NU(1))*F10M1+Z*F100	00025950
1615*	GO TO 1190	00025960
1616*	1080 INT(K) = -2.0*(1.0-NU(1))*F110+Z*F111	00025970
1617*	GO TO 1190	00025980
1618*	1090 INT(K) = -F110	00025990
1619*	GO TO 1190	00026000
1620*	1100 INT(K) = F11M1-Z*F110	00026010
1621*	GO TO 1180	00026020
1622*	1110 INT(K) = F11M1	00026030
1623*	GO TO 1180	00026040
1624*	1120 INT(K) = -2.0*(1.0-NU(1))*F11M2+Z*F11M1	00026050
1625*	GO TO 1180	00026060
1626*	1130 INT(K) = -F100	00026070
1627*	GO TO 1190	00026080
1628*	1140 INT(K) = F10M1	00026090
1629*	GO TO 1190	00026100
1630*	1150 INT(K) = F110	00026110
1631*	GO TO 1190	00026120
1632*	1160 INT(K) = -F11M1	00026130
1633*	GO TO 1180	00026140
1634*	1170 INT(K) = F11M2	00026150
1635*	1180 IF(R.GE.ACCUR(1)) INT(K) = INT(K)/R	00026160
1636*	1190 CONTINUE	00026170
1637*	IF(NLAYS.EQ.1) GO TO 3140	00026180
1638*	C-----	00026190
1639*	C	00026200
1640*	C	00026210
1641*	C-----	00026220

1642*	2000	INTT2 = INTT	00026230
1643*		INTT3 = INTT	00026240
1644*		DO 2010 J = 1,NINT	00026250
1645*		INTV2(J) = INTV(J)	00026260
1646*		INTV3(J) = INTV(J)	00026270
1647*	2010	CONTINUE	00026280
1648*		UPPER = ZEROS(1)	00026290
1649*		ALFA = 0	00026300
1650*		BETA = 0	00026310
1651*		IRFS = 0	00026320
1652*		DELTA = 0.5*ZEROS(1)	00026330
1653*	2020	LOWER = UPPER-DELTA	00026340
1654*		IF(LOWER-ACCUR(1)) 2030,2030,2040	00026350
1655*	2030	ALFA = -1	00026360
1656*		LOWER = 0.0	00026370
1657*	2040	IF(IRES.EQ.1) GO TO 2050	00026380
1658*		CALL QUAD(IL,INTV3,LOWER,UPPER,16,COMP,NTELL)	00026390
1659*		IF(NTELL.NE.0) GO TO 3100	00026400
1660*		GO TO 2070	00026410
1661*	2050	DO 2060 J = 1,NINT	00026420
1662*		COMP(J) = RES(J)	00026430
1663*	2060	CONTINUE	00026440
1664*		IRES = 0	00026450
1665*	2070	MIDPNT = 0.5*(LOWER+UPPER)	00026460
1666*		CALL QUAD(IL,INTV3,LOWER,MIDPNT,16,FIRST,NTELL)	00026470
1667*		IF(NTELL.NE.0) GO TO 3100	00026480
1668*		CALL QUAD(IL,INTV3,MIDPNT,UPPER,16,SECOND,NTELL)	00026490
1669*		IF(NTELL.NE.0) GO TO 3100	00026500
1670*		DO 2090 J = 1,NINT	00026510
1671*		IF(INTV3(J).EQ.0) GO TO 2090	00026520
1672*		IF(ABS(COMP(J)).LT.ACCUR(2)) GO TO 2080	00026530
1673*		IF(ABS((COMP(J)-FIRST(J)-SECOND(J))/COMP(J)).LT.ACCUR(3))	00026540
1674*	1	GO TO 2080	00026550
1675*		GO TO 2090	00026560
1676*	2080	INTT3 = INTT3-1	00026570
1677*		IF(LOWER.GT.ACCUR(1)) GO TO 2090	00026580
1678*		INTT2 = INTT2-1	00026590
1679*		INTV3(J) = 0	00026600
1680*	2090	CONTINUE	00026610
1681*		IF(INTT3.EQ.0) GO TO 2110	00026620
1682*		ALFA = 0	00026630
1683*		LOWER = MIDPNT	00026640
1684*		DELTA = 0.5*DELTA	00026650
1685*		BETA = BETA+1	00026660
1686*		IF(BETA.GT.30) GO TO 2150	00026670
1687*	C-----		00026680
1688*	C	ARRIVAL HERE MEANS THAT THE INTEGRAND IS TOO	00026690
1689*	C	IRREGULAR TO GET INTEGRATED OVER THE REGION FROM	00026700
1690*	C	THE ORIGIN TO THE FIRST BESSEL ZERO.	00026710
1691*	C-----		00026720
1692*		IRES = 1	00026730
1693*		DO 2100 J = 1,NINT	00026740
1694*		COMP(J) = SECOND(J)	00026750

1695*		RES(J) = FIRST(J)	00026760
1696*	2100	CONTINUE	00026770
1697*		INTT3 = INTT2	00026780
1698*		GO TO 2070	00026790
1699*	2110	DO 2120 J = 1,NINT	00026800
1700*		K = INTV2(J)	00026810
1701*		IF(K.EQ.0) GO TO 2120	00026820
1702*		INT(K) = INT(K)+FIRST(J)+SECOND(J)	00026830
1703*		IF(INTV3(J).NE.0) GO TO 2120	00026840
1704*		INTV2(J) = 0	00026850
1705*	2120	CONTINUE	00026860
1706*		UPPER = LOWER	00026870
1707*		INTT3 = INTT2	00026880
1708*		IF(ALFA) 3000,2140,2130	00026890
1709*	2130	DELTA = DELTA*2.0	00026900
1710*		BETA = BETA-1	00026910
1711*	2140	ALFA = ALFA+1	00026920
1712*		GO TO 2020	00026930
1713*	2150	WRITE(NDUT,9040)	00026940
1714*		GO TO 3180	00026950
1715*	C-----	-----	00026960
1716*	C	INTEGRATION OVER THE REMAINING INTERVALS.	00026970
1717*	C-----	-----	00026980
1718*	3000	IFIN = NZEROS-1	00026990
1719*		DO 3010 J = 1,NINT	00027000
1720*		INTV2(J) = INTV(J)	00027010
1721*	3010	CONTINUE	00027020
1722*		INTT2 = INTT	00027030
1723*		DO 3130 IBESS = 1,IFIN	00027040
1724*		DO 3020 J = 1,NINT	00027050
1725*		INTV3(J) = INTV2(J)	00027060
1726*		FIRST(J) = 0.0	00027070
1727*	3020	CONTINUE	00027080
1728*		INTT3 = INTT2	00027090
1729*		DO 3070 ORDER = 4,15	00027100
1730*		CALL QUAD(IL,INTV3,ZEROS(IBESS),ZEROS(IBESS+1),ORDER,SECOND,	00027110
1731*	1	NTELL)	00027120
1732*		IF(NTELL.NE.0) GO TO 3180	00027130
1733*		DO 3060 J = 1,NINT	00027140
1734*		K = INTV3(J)	00027150
1735*		IF(K.EQ.0) GO TO 3060	00027160
1736*		IF(ABS(INT(K)).LT.0.01) GO TO 3030	00027170
1737*		IF(ABS((FIRST(J)-SECOND(J))/INT(K)).LT.0.1*ACCUR(3))	00027180
1738*	1	GO TO 3040	00027190
1739*		GO TO 3050	00027200
1740*	3030	IF(ABS(FIRST(J)-SECOND(J)).GE.0.1*ACCUR(2)) GO TO 3050	00027210
1741*	3040	INTV3(J) = 0	00027220
1742*		INTT3 = INTT3-1	00027230
1743*		GO TO 3060	00027240
1744*	3050	FIRST(J) = SECOND(J)	00027250
1745*	3060	CONTINUE	00027260
1746*		IF(INTT3.EQ.0) GO TO 3080	00027270
1747*	3070	CONTINUE	00027280

1748*	WRITE(NOUT,9020)	00027290
1749*	WRITE(NOUT,9050) ZEROS(IBESS)	00027300
1750*	C-----	00027310
1751*	C ARRIVAL HERE MEANS THAT THE DESIRED ACCURACY CANNOT	00027320
1752*	C BE MET BY MEANS OF THE AVAILABLE GAUSS-JACOBI	00027330
1753*	C POLYNOMIALS.	00027340
1754*	C-----	00027350
1755*	GO TO 3180	00027360
1756*	3080 DO 3120 J = 1,NINT	00027370
1757*	K = INTV2(J)	00027380
1758*	IF(K.EQ.0) GO TO 3120	00027390
1759*	INT(K) = INT(K)+SECOND(J)	00027400
1760*	IF(ABS(INT(K)).LT.0.01) GO TO 3090	00027410
1761*	IF(ABS(SECOND(J)/INT(K)).LT.0.1*ACCUR(3)) GO TO 3110	00027420
1762*	GO TO 3100	00027430
1763*	3090 IF(ABS(SECOND(J)).LT.0.1*ACCUR(2)) GO TO 3110	00027440
1764*	3100 KK(J) = 0	00027450
1765*	GO TO 3120	00027460
1766*	3110 KK(J) = KK(J)+1	00027470
1767*	IF(KK(J).LT.2) GO TO 3120	00027480
1768*	INTV2(J) = 0	00027490
1769*	INTT2 = INTT2-1	00027500
1770*	3120 CONTINUE	00027510
1771*	IF(INTT2.EQ.0) GO TO 3140	00027520
1772*	3130 CONTINUE	00027530
1773*	WRITE(NOUT,9030)	00027540
1774*	WRITE(NOUT,9050) ZEROS(IFIN)	00027550
1775*	C-----	00027560
1776*	C ARRIVAL HERE MEANS THAT ALL AVAILABLE BESSEL ZEROS	00027570
1777*	C HAVE BEEN EXHAUSTED BECAUSE OF ILL CONVERGENCE OF	00027580
1778*	C THE INTEGRALS.	00027590
1779*	C-----	00027600
1780*	GO TO 3180	00027610
1781*	3140 DO 3170 J = 1,NINT	00027620
1782*	K = INTV(J)	00027630
1783*	IF(K.EQ.0) GO TO 3170	00027640
1784*	IF(K-5) 3150,3150,3160	00027650
1785*	3150 INT(K) = INT(K)*LOAD	00027660
1786*	GO TO 3170	00027670
1787*	3160 INT(K) = INT(K)*HOSTRS	00027680
1788*	3170 CONTINUE	00027690
1789*	RETURN	00027700
1790*	3180 WRITE(NOUT,9010) (INTV3(J),J=1,NINT)	00027710
1791*	GO TO 3140	00027720
1792*	9010 FORMAT(' DURING CALCULATION OF INTEGRALS',10I3)	00027730
1793*	9020 FORMAT(' SUSPEND PROGRAM GAUSS POLYS EXHAUSTED')	00027740
1794*	9030 FORMAT(' SUSPEND PROGRAM BESSEL ZEROS EXHAUSTED')	00027750
1795*	9040 FORMAT(' SUSPEND PROGRAM STEPSIZE FIRST INTERVAL TOO SMALL')	00027760
1796*	9050 FORMAT(' AT THE VALUE #',E11.4,' FOR THE INTEGRATION VARIABLE')	00027770
1797*	END	00027780
1798*	SUBROUTINE QUAD (IL,INTV,ALO,UP,NGAUSS,FSC,NTLL)	00027790
1799*	C-----	00027800
1800*	C THIS SUBROUTINE CALCULATES FOR THE SET	00027810

1801* C	INTV THE INTEGRALS OF THE CORRESPONDING	00027820
1802* C	FUNCTIONS IGRAND BETWEEN THE LIMITS ALO	00027830
1803* C	AND UP BY USING A GAUSS QUADRATURE OF	00027840
1804* C	ORDER NGAUSS.	00027850
1805* C	-FOR NGAUSS=16 A LEGENDRE-GAUSS QUADRATU-	00027860
1806* C	RE OF ORDER 8.	00027870
1807* C	-FOR NGAUSS.LT.16 A JACOBI-GAUSSQUADRATU-	00027880
1808* C	RE.	00027890
1809* C	THE ABSCISSAE AND WEIGHTS OF BOTH ARE	00027900
1810* C	STORED AS AGAUSS AND HGAUSS IN THE BLOCK	00027910
1811* C	DATA.	00027920
1812* C	THE SET OF INTEGRANDS IS COMPUTED DURING	00027930
1813* C	SUBSEQUENT CALLING IN OF#	00027940
1814* C	SUBROUTINES MATRIX	00027950
1815* C	FPIGRA	00027960
1816* C	AND FUNCTION IGRAND	00027970
1817* C	THE SET OF RESULTING INTEGRALS IS	00027980
1818* C	DELIVERED IN FSC	00027990
1819* C	-----	00028000
1820*	INTEGER INTV(10)	00028010
1821*	REAL IGRAND,FSC(10)	00028020
1822*	COMMON/GAUSS/AGAUSS(16,16),HGAUSS(16,16)	00028030
1823*	NINT = 7	00028040
1824*	IF(IL.EQ.2) NINT = 10	00028050
1825*	DO 10 J = 1,NINT	00028060
1826*	K = INTV(J)	00028070
1827*	IF(K.EQ.0) GO TO 10	00028080
1828*	FSC(J) = 0.0	00028090
1829*	10 CONTINUE	00028100
1830*	LABEL = 0	00028110
1831*	IF(IL.EQ.2) GO TO 20	00028120
1832*	IF((INTV(1)+INTV(2)+INTV(3)).GT.0) LABEL = LABEL+1	00028130
1833*	IF((INTV(4)+INTV(5)).GT.0) LABEL = LABEL+2	00028140
1834*	IF((INTV(6)+INTV(7)).GT.0) LABEL = LABEL+4	00028150
1835*	GO TO 30	00028160
1836*	20 IF((INTV(1)+INTV(2)).GT.0) LABEL = LABEL+1	00028170
1837*	IF((INTV(3)+INTV(4)+INTV(5)+INTV(6)+INTV(7)).GT.0) LABEL = LABEL+2	00028180
1838*	IF((INTV(8)+INTV(9)+INTV(10)).GT.0) LABEL = LABEL+4	00028190
1839*	30 F1 = 0.5*(UP-ALO)	00028200
1840*	F2 = 0.5*(UP+ALO)	00028210
1841*	IGAUSS = NGAUSS	00028220
1842*	IF(NGAUSS.EQ.16) IGAUSS=8	00028230
1843*	DO 50 I = 1,IGAUSS	00028240
1844*	X = F1*AGAUSS(I,NGAUSS)+F2	00028250
1845*	CALL MATRIX (X,LABEL,NTELL)	00028260
1846*	IF(NTELL.EQ.1) RETURN	00028270
1847*	CALL FPIGRA (IL,X)	00028280
1848*	DO 40 J = 1,NINT	00028290
1849*	K = INTV(J)	00028300
1850*	IF(K.EQ.0) GO TO 40	00028310
1851*	FSC(J) = FSC(J)+HGAUSS(I,NGAUSS)*IGRAND(X,K)	00028320
1852*	40 CONTINUE	00028330
1853*	50 CONTINUE	00028340

```

1854*      DO 60 J = 1,NINT                      00028350
1855*      K = INTV(J)                          00028360
1856*      IF(K.EQ.0) GO TO 60                  00028370
1857*      FSC(J) = FSC(J)*F1                    00028380
1858*      60 CONTINUE                          00028390
1859*      70 RETURN                            00028400
1860*      END                                  00028410
1861*      SUBROUTINE MATRIX (X,LABL,NTELL)      00028420
1862* C-----                                00028430
1863* C      THIS SUBROUTINE COMPUTES THE SET OF CHA- 00028440
1864* C      RACTERISTIC-FUNCTIONS TO,V0,S0,U0,T1,V1, 00028450
1865* C      S1,U1,TQ1 AND SQ1 FOR THE VALUE X OF THE 00028460
1866* C      INTEGRATION-PARAMETER. USE IS MADE OF    00028470
1867* C      CONSTANTS CALCULATED IN MACON1 AND MA2CON. 00028480
1868* C      THEY WERE STORED IN COMMON/INDATA/.      00028490
1869* C      CHARACTERISTIC-FUNCTION VALUES ARE DELIVE-00028500
1870* C      RED IN COMMON/IGRAN/.                    00028510
1871* C      LABL DETERMINES WHICH CHARACTERISTIC-    00028520
1872* C      FUNCTIONS ARE NEEDED#                     00028530
1873* C      -LABL=1#T0,V0,S0,U0                      00028540
1874* C      -LABL=2#T1,V1,S1,U1                      00028550
1875* C      -LABL=3#T0,V0,S0,U0,T1,V1,S1,U1        00028560
1876* C      -LABL=4#TQ1,SQ1                         00028570
1877* C      -LABL=5#T0,V0,S0,U0,TQ1,SQ1              00028580
1878* C      -LABL=6#T1,V1,S1,U1,TQ1,SQ1             00028590
1879* C      -LABL=7#T0,V0,S0,U0,T1,V1,S1,U1,TQ1,SQ1 00028600
1880* C      SUBROUTINE IS INTERRUPTED AND RETURNED   00028610
1881* C      WITH NTELL=1 WHEN SOLUTION BECOMES TOO   00028620
1882* C      INACCURATE BECAUSE OF ILL MATRIX-CONDI- 00028630
1883* C      ON DURING INVERSION.                     00028640
1884* C-----                                00028650
1885*      REAL LOAD,NU,W(4,4,9),P(4,2),PP(2,2),K1,K2,K5,K6,II,NJ(2,2,9),KK6, 00028660
1886*      1ACCUR(3),NP(2,10),NJ2(9),P3(2),NP2(10),K4(10) 00028670
1887*      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9) 00028680
1888*      1,K5(10),E(10),AL(9),THICK(9),RADIUS(10)        00028690
1889*      COMMON/INDATA/XMAX, A1(9),B1(9),C1(9),D(9),EE(9),F(9),G(9),H1(9), 00028700
1890*      1II(9),K1(9),K2(9),K6(10),BE(9),BU(9),BUU(9),BMU(9),B2U(9),B2UU(9), 00028710
1891*      2J2(9),J1,T2(10),SS(2,10),G012(9),G021(9),G022(9),G122(9), 00028720
1892*      3H012(9),H022(9),H122(9),D012(9),D022(9),C011(9),C012(9),E012(9), 00028730
1893*      4F012(9),F112(9),F022(9),CC(4,2,9),DD(2,2,9),FF(2,2,9),GG(2,2,10), 00028740
1894*      5HH(2,2,10),RR(4,2,10),DD2(9),G20(9),G21(9),H20(9),H021(9),GG2(10), 00028750
1895*      6HH2(10),Q011(9),Q111(9),Q012(9),Q112(9),Q212(9),Q022(9),Q122(9), 00028760
1896*      7QF0(9),QF1(9),Z011,Z111,Z211,Z012,Z112,Z212,Z312,Z021,Z121,Z022, 00028770
1897*      8Z122,Z222,K4 00028780
1898*      COMMON/IGRAN/T0,V0,S0,U0,T1,V1,S1,U1,TQ1,SQ1,FP1GR,EX1,EX2 00028790
1899*      COMMON/TAPE/NOUT 00028800
1900*      LABEL = LABL 00028810
1901*      IF(LABEL.LT.4) GO TO 1000 00028820
1902*      IF(X.LT.XMAX) GO TO 100 00028830
1903* C-----                                00028840
1904* C      ASYMPTOTIC EVALUATION OF TQ1 AND SQ1      00028850
1905* C      FOR X.GE.XMAX. 00028860
1906* C-----                                00028870

```

1907*	TQ1 = 1.0	00028880
1908*	IF(LAYER.EQ.1) GO TO 30	00028890
1909*	J = LAYER-1	00028900
1910*	DO 20 K = 1,J	00028910
1911*	TQ1=TQ1*2.0*(1.0-AL(K))/((1.0-AL(K))*(1.0+K1(K))+0.5*AL(K)*X)	00028920
1912*	20 CONTINUE	00028930
1913*	30 SQ1=TQ1*(0.5*AL(LAYER)*X-(1.0-AL(LAYER))*K2(LAYER))/((1.0-AL(LAYER00028940	
1914*	1))*(1.0+K1(LAYER))+0.5*AL(LAYER)*X)	00028950
1915*	LABEL = LABEL-4	00028960
1916*	GO TO 1000	00028970
1917*	C-----	00028980
1918*	C CALCULATION OF TQ1 AND SQ1 FOR X.LT.XMAX	00028990
1919*	C-----	00029000
1920*	100 IF(J1.EQ.0) GO TO 120	00029010
1921*	DO 110 J = 1,J1	00029020
1922*	GG2(J+1) = G20(J)-G21(J)*X	00029030
1923*	HH2(J+1) = H20(J)+G21(J)*X	00029040
1924*	110 CONTINUE	00029050
1925*	120 DO 150 K = 1,M	00029060
1926*	IF(J1.EQ.0) GO TO 140	00029070
1927*	DO 130 I = 1,J1	00029080
1928*	IF(J2(I).EQ.K) GO TO 150	00029090
1929*	130 CONTINUE	00029100
1930*	140 W1 = 0.5*(1.0+K1(K))	00029110
1931*	W2 = -0.5*K2(K)	00029120
1932*	W3 = BE(K)*0.25*X	00029130
1933*	NJ(1,1,K) = W1+W3	00029140
1934*	NJ(1,2,K) = W2-W3	00029150
1935*	NJ(2,1,K) = W2+W3	00029160
1936*	NJ(2,2,K) = W1-W3	00029170
1937*	150 CONTINUE	00029180
1938*	J5 = J1+1	00029190
1939*	DO 300 MM = 1,J5	00029200
1940*	N = J5+1-MM	00029210
1941*	IF(N-1) 160,160,170	00029220
1942*	160 J3 = 1	00029230
1943*	GO TO 180	00029240
1944*	170 J3 = J2(N-1)+1	00029250
1945*	180 IF(J5-N) 190,190,200	00029260
1946*	190 J4 = M	00029270
1947*	GO TO 210	00029280
1948*	200 J4 = J2(N)-1	00029290
1949*	210 IF(J3.GT.J4) GO TO 240	00029300
1950*	DO 230 IJ = J3,J4	00029310
1951*	IK = J4+J3-IJ	00029320
1952*	IL = IK+1	00029330
1953*	EXPO=-X*MT2(IL)	00029340
1954*	IF(EXPO.LT.-70.0)GO TO 212	00029350
1955*	EXP1=EXP(EXPO)*SS(1,IL)	00029360
1956*	GO TO 214	00029370
1957*	212 EXP1=0.0	00029380
1958*	214 DO 220 I=1,2	00029390
1959*	220 SS(I,IK) = NJ(I,1,IK)*EXP1+NJ(I,2,IK)*SS(2,IL)	00029400

1960*	230	CONTINUE	000294.0
1961*	240	NN = N-1	00029420
1962*		EXP0=-X*T2(J3)	00029430
1963*		IF(EXP0.LT.-70.0)GO TO 242	00029440
1964*		EXP2=EXP(EXP0)	00029450
1965*		GO TO 244	00029460
1966*	242	EXP2=0.0	00029470
1967*	244	PROD=GG2(N)*SS(1,J3)*EXP2	00029480
1968*		P2 = PROD+HH2(N)*SS(2,J3)	00029490
1969*	C	-----	00029500
1970*	C	TEST MATRIX-CONDITION BEFORE INVERSION.	00029510
1971*	C	-----	00029520
1972*		IF(ABS(P2).LT.1.0E-7*ABS(PROD)/ACCUR(3)) GO TO 2000	00029530
1973*		PP2 = 1.0/P2	00029540
1974*		IF(N.EQ.1) GO TO 310	00029550
1975*		NJ2(NN) = PP2*DD2(NN)	00029560
1976*		DO 270 I = 1,2	00029570
1977*	270	P3(I) = SS(1,J3)*NJ2(NN)	00029580
1978*		PP2=P3(1)*EXP2-P3(2)	00029590
1979*		SS(1,J3-1) = PP2+1.0	00029600
1980*		SS(2,J3-1) = 1.0	00029610
1981*	300	CONTINUE	00029620
1982*	310	IF(NTELL.EQ.2) RETURN	00029630
1983*		NP2(1) = -PP2	00029640
1984*		IF(J1.EQ.0) GO TO 390	00029650
1985*		DO 350 I = 1,J1	00029660
1986*		J = J1+1-I	00029670
1987*		IF(LAYER.GT.J2(J)) GO TO 360	00029680
1988*	350	CONTINUE	00029690
1989*		J5 = 1	00029700
1990*		GO TO 390	00029710
1991*	360	DO 380 I = 1,J	00029720
1992*		NP2(I+1) = NJ2(I)*NP2(I)	00029730
1993*	380	CONTINUE	00029740
1994*		J5 = J+1	00029750
1995*	390	J = LAYER	00029760
1996*		SQ1 = SS(1,J)*NP2(J5)	00029770
1997*		TQ1 = SS(2,J)*NP2(J5)	00029780
1998*		LABEL = LABEL-4	00029790
1999*	C	-----	00029800
2000*	C	ASYMPTOTIC EVALUATION OF T0,V0,S0,U0,T1,	00029810
2001*	C	V1,S1 AND U1 FOR X.GE.XMAX.	00029820
2002*	C	-----	00029830
2003*	1000	IF(LABEL.EQ.0) RETURN	00029840
2004*		IF(X.LT.XMAX) GO TO 1100	00029850
2005*		L = LAYER	00029860
2006*		X2 = X*X	00029870
2007*		X3 = X2*X	00029880
2008*		IF(L.EQ.NLAYS) GO TO 1010	00029890
2009*		Z11 = Z011+X*Z111+X2*Z211	00029900
2010*		Z12 = Z012+X*Z112+X2*Z212+X3*Z312	00029910
2011*		Z21 = Z021+X*Z121	00029920
2012*		Z22 = Z022+X*Z122+X2*Z222	00029930

2013*	1010	IF(LABEL.GT.1) GO TO 1030	00029940
2014*	1020	IF(LABEL.EQ.0) RETURN	00029950
2015*		NP(1,1) = 2.0*NU(1)	00029960
2016*		NP(2,1) = 1.0	00029970
2017*		GO TO 1040	00029980
2018*	1030	NP(1,1) = 1.0-2.0*NU(1)	00029990
2019*		NP(2,1) = -1.0	00030000
2020*	1040	PQF = 1.0	00030010
2021*		IF(L.EQ.1) GO TO 1060	00030020
2022*		DO 1050 K = 2,1	00030030
2023*		J = K-1	00030040
2024*		PQF = PQF*K6(J)/(QF0(J)+QF1(J)*X)	00030050
2025*		W1 = -AL(J)*X	00030060
2026*		W9 = H(J)*X	00030070
2027*		NP(1,K) = NP(1,J)*(Q011(J)+Q111(J)*X+W1*W9)+NP(2,J)*(Q012(J)	00030080
2028*	1	+Q112(J)*X+Q212(J)*X2+W1*W9*W9)	00030090
2029*	1050	NP(2,K) = -W1*NP(1,J)+NP(2,J)*(Q022(J)+Q122(J)*X-W1*W9)	00030100
2030*		IF(L.NE.NLAYS) GO TO 1060	00030110
2031*		S = 0.0	00030120
2032*		U = 0.0	00030130
2033*		GO TO 1070	00030140
2034*	1060	S = (NP(1,L)*Z11+NP(2,L)*Z12)*PQF/(QF0(L)+QF1(L)*X)	00030150
2035*		U = (NP(1,L)*Z21+NP(2,L)*Z22)*PQF/(QF0(L)+QF1(L)*X)	00030160
2036*	1070	T = NP(1,L)*PQF	00030170
2037*		V = NP(2,L)*PQF	00030180
2038*		IF(LABEL.GT.1) GO TO 1080	00030190
2039*		S0 = S	00030200
2040*		U0 = U	00030210
2041*		T0 = T	00030220
2042*		V0 = V	00030230
2043*		RETURN	00030240
2044*	1080	S1 = S	00030250
2045*		U1 = U	00030260
2046*		T1 = T	00030270
2047*		V1 = V	00030280
2048*		LABEL = LABEL-2	00030290
2049*		GO TO 1020	00030300
2050*	C	-----	00030310
2051*	C	CALCULATION OF T0,V0,S0,U0,T1,V1,S1 AND	00030320
2052*	C	U1 FOR X.LT.XMAX.	00030330
2053*	C	-----	00030340
2054*	1100	IF(J1.EQ.0) GO TO 1120	00030350
2055*		DO 1110 J = 1,J1	00030360
2056*		K = J2(J)	00030370
2057*		W1 = -AL(K)*X	00030380
2058*		W9 = H(K)*X	00030390
2059*		CC(1,1,J) = C011(J)+2.0*W9	00030400
2060*		CC(1,2,J) = C012(J)+2.0*W9*W9	00030410
2061*		CC(2,2,J) = C011(J)-2.0*W9	00030420
2062*		DD(1,2,J) = D012(J)+DD(1,1,J)*W9	00030430
2063*		DD(2,2,J) = D022(J)+DD(2,1,J)*W9	00030440
2064*		FF(1,1,J) = -C011(J)-2.0*W9	00030450
2065*		FF(1,2,J) = F012(J)+F112(J)*W9-2.0*W9*W9	00030460

2066*		FF(2,2,J) = F022(J)+2.0*W9	00030470
2067*		GG(1,2,J+1) = G012(J)+GG(1,1,J+1)*W9	00030480
2068*		GG(2,1,J+1) = G021(J)+W1	00030490
2069*		GG(2,2,J+1) = G022(J)+(G021(J)*H(K)+G122(J))*X+W1*W9	00030500
2070*		HH(1,2,J+1) = H012(J)+HH(1,1,J+1)*W9	00030510
2071*		HH(2,1,J+1) = H021(J)+W1	00030520
2072*		HH(2,2,J+1) = H022(J)+H021(J)*W9+H122(J)*X+W1*W9	00030530
2073*	1110	CONTINUE	00030540
2074*	1120	DO 1150 K=1,M	00030550
2075*		IF(J1.EQ.0) GO TO 1140	00030560
2076*		DO 1130 I = 1,J1	00030570
2077*		IF(J2(I).EQ.K) GO TO 1150	00030580
2078*	1130	CONTINUE	00030590
2079*	1140	W1 = BMU(K)*X	00030600
2080*		W9 = H(K)*X	00030610
2081*		W10 = W9*X	00030620
2082*		W2 = W10*BE(K)	00030630
2083*		W11 = W2*W9	00030640
2084*		W3 = W9*C1(K)	00030650
2085*		W4 = BE(K)*X	00030660
2086*		W5 = BU(K)*X	00030670
2087*		W8 = BUU(K)*X	00030680
2088*		W7 = C1(K)*W9*W9	00030690
2089*		W(1,1,K) = A1(K)+W1-W2	00030700
2090*		W(1,2,K) = -EE(K)+F(K)*W9+W8+B2U(K)*W10-W11	00030710
2091*		W(1,3,K) = D(K)-W3+W1-W2	00030720
2092*		W(1,4,K) = -G(K)+H1(K)*W9-BUU(K)*X-W7+B2UU(K)*W10-W11	00030730
2093*		W(2,1,K) = W4	00030740
2094*		W(2,2,K) = B1(K)+W5+W2	00030750
2095*		W(2,3,K) = C1(K)+W4	00030760
2096*		W(2,4,K) = II(K)+W3-W5+W2	00030770
2097*		W(3,1,K) = D(K)+W3-W1-W2	00030780
2098*		W(3,2,K) = G(K)+H1(K)*W9-W8+W7-B2UU(K)*W10-W11	00030790
2099*		W(3,3,K) = A1(K)-W1-W2	00030800
2100*		W(3,4,K) = EE(K)+F(K)*W9+W8-B2U(K)*W10-W11	00030810
2101*		W(4,1,K) = -C1(K)+W4	00030820
2102*		W(4,2,K) = II(K)-W3+W5+W2	00030830
2103*		W(4,3,K) = W4	00030840
2104*		W(4,4,K) = B1(K)-W5+W2	00030850
2105*	1150	CONTINUE	00030860
2106*		J5 = J1+1	00030870
2107*		PKK6 = 1.0	00030880
2108*		DO 1300 MM = 1,J5	00030890
2109*		KK6 = 1.0	00030900
2110*		N = J5+1-MM	00030910
2111*		IF(N-1) 1160,1160,1170	00030920
2112*	1160	J3 = 1	00030930
2113*		GO TO 1180	00030940
2114*	1170	J3 = J2(N-1)+1	00030950
2115*	1180	IF(J5-N) 1190,1190,1200	00030960
2116*	1190	J4 = M	00030970
2117*		GO TO 1210	00030980
2118*	1200	J4 = J2(N)-1	00030990


```

2119* 1210 IF(J3.GT.J4) GO TO 1240 00031000
2120* DO 1230 IJ = J3,J4 00031010
2121* IK = J4+J3-IJ 00031020
2122* IL = IK+1 00031030
2123* KK6 = KK6*KK6(IK) 00031040
2124* IF(IK.EQ.LAYER) PKK6 = KK6 00031050
2125* EXPD=-X*T2(IL) 00031060
2126* IF(EXPD.LT.-70.0)GO TO 1212 00031070
2127* EXP1=EXP(EXPD) 00031080
2128* GO TO 1214 00031090
2129* 1212 EXP1=0.0 00031100
2130* 1214 DO 1220 I=1,4 00031110
2131* DO 1220 K = 1,2 00031120
2132* 1220 RR(I,K,IK)=(W(I,1,IK)*RR(1,K,IL)+W(I,2,IK)*RR(2,K,IL)) 00031130
2133* 1 *EXP1+W(I,3,IK)*RR(3,K,IL)+W(I,4,IK)*RR(4,K,IL) 00031140
2134* 1230 CONTINUE 00031150
2135* 1240 NN = N-1 00031160
2136* EXPD=-X*T2(J3) 00031170
2137* IF(EXPD.LT.-70.0)GO TO 1242 00031180
2138* EXP2=EXP(EXPD) 00031190
2139* GO TO 1244 00031200
2140* 1242 EXP2=0.0 00031210
2141* 1244 DO 1250 I=1,2 00031220
2142* DO 1250 K = 1,2 00031230
2143* 1250 P(I,K) = (GG(I,1,N)*RR(1,K,J3)+GG(I,2,N)*RR(2,K,J3)) 00031240
2144* 1 *EXP2+HH(I,1,N)*RR(3,K,J3)+HH(I,2,N)*RR(4,K,J3) 00031250
2145* PROD = P(1,1)*P(2,2) 00031260
2146* DET = PROD-P(1,2)*P(2,1) 00031270
2147* C----- 00031280
2148* C TEST MATRIX.CONDITION BEFORE INVERSION. 00031290
2149* C----- 00031300
2150* IF(ABS(DET).LT.1.0E-7*ABS(PROD)/ACCUR(3)) GO TO 2000 00031310
2151* QKK6 = KK6/DET 00031320
2152* PP(1,1) = P(2,2)*QKK6 00031330
2153* PP(1,2) = -P(1,2)*QKK6 00031340
2154* PP(2,1) = -P(2,1)*QKK6 00031350
2155* PP(2,2) = P(1,1)*QKK6 00031360
2156* IF(N.EQ.1) GO TO 1310 00031370
2157* DO 1260 I=1,2 00031380
2158* DO 1260 K = 1,2 00031390
2159* 1260 NJ(I,K,NN)=PP(I,1)*DD(1,K,NN)+PP(I,2)*DD(2,K,NN) 00031400
2160* DO 1270 I = 1,4 00031410
2161* DO 1270 K = 1,2 00031420
2162* 1270 P(I,K) =(RR(I,1,J3)*NJ(1,K,NN)+RR(I,2,J3)*NJ(2,K,NN))/KK6 00031430
2163* DO 1280 I = 1,2 00031440
2164* PP(1,I)=(P(1,I)+E012(NN)*P(2,I))*EXP2+FF(1,1,NN) 00031450
2165* 1 *P(3,I)+FF(1,2,NN)*P(4,I) 00031460
2166* 1280 PP(2,I)=P(2,I)*EXP2+FF(2,1,NN)*P(3,I)+FF(2,2,NN)*P(4,I) 00031470
2167* DO 1290 I = 1,2 00031480
2168* DO 1290 K = 1,2 00031490
2169* 1290 RR(I,K,J3-1) = CC(I,K,NN)+PP(I,K) 00031500
2170* RR(3,1,J3-1) = 1.0 00031510
2171* RR(3,2,J3-1) = 0.0 00031520

```

2172*	RR(4,1,J3-1) = 0.0	00031530
2173*	RR(4,2,J3-1) = 1.0	00031540
2174*	1300 CONTINUE	00031550
2175*	1310 IF(NTELL.EQ.2) GO TO 100	00031560
2176*	IF(LABEL.GT.1) GO TO 1330	00031570
2177*	1320 IF(LABEL.EQ.0) RETURN	00031580
2178*	NP(1,1) = PP(1,1)	00031590
2179*	NP(2,1) = PP(2,1)	00031600
2180*	GO TO 1340	00031610
2181*	1330 NP(1,1) = PP(1,2)	00031620
2182*	NP(2,1) = PP(2,2)	00031630
2183*	1340 IF(J1.EQ.0) GO TO 1390	00031640
2184*	DO 1350 I = 1,J1	00031650
2185*	J = J1+1-I	00031660
2186*	IF(LAYER.GT.J2(J)) GO TO 1360	00031670
2187*	1350 CONTINUE	00031680
2188*	J5 = 1	00031690
2189*	GO TO 1390	00031700
2190*	1360 DO 1380 I = 1,J	00031710
2191*	IH = I+1	00031720
2192*	DO 1370 K = 1,2	00031730
2193*	1370 NP(K,IH) = NJ(K,1,I)*NP(1,I)+NJ(K,2,I)*NP(2,I)	00031740
2194*	1380 CONTINUE	00031750
2195*	J5 = J+1	00031760
2196*	1390 J = LAYER	00031770
2197*	S = (RR(1,1,J)*NP(1,J5)+RR(1,2,J)*NP(2,J5))/PKK6	00031780
2198*	U = (RR(2,1,J)*NP(1,J5)+RR(2,2,J)*NP(2,J5))/PKK6	00031790
2199*	T = (RR(3,1,J)*NP(1,J5)+RR(3,2,J)*NP(2,J5))/PKK6	00031800
2200*	V = (RR(4,1,J)*NP(1,J5)+RR(4,2,J)*NP(2,J5))/PKK6	00031810
2201*	IF(LABEL.GT.1) GO TO 1400	00031820
2202*	T0 = T	00031830
2203*	S0 = S	00031840
2204*	U0 = U	00031850
2205*	V0 = V	00031860
2206*	RETURN	00031870
2207*	1400 S1 = S	00031880
2208*	T1 = T	00031890
2209*	U1 = U	00031900
2210*	V1 = V	00031910
2211*	LABEL = LABEL-2	00031920
2212*	GO TO 1320	00031930
2213*	C-----	00031940
2214*	C ARRIVAL HERE MEANS THAT SOLUTION OF THE	00031950
2215*	C CHARACTERISTIC FUNCTIONS HAS BEEN STOPPED	00031960
2216*	C PREMATURELY BECAUSE OF ILL MATRIX CONDI-	00031970
2217*	C TION MET DURING SOLUTION PROCESS.	00031980
2218*	C-----	00031990
2219*	2000 WRITE(NOUT,9000)X	00032000
2220*	NTELL = 1	00032010
2221*	RETURN	00032020
2222*	9000 FORMAT(' ILL-CONDITIONED DETERMINANT FOR X=',E15.7)	00032030
2223*	END	00032040
2224*	SUBROUTINE FPIGRA (IL,X)	00032050

```

2225* C-----00032060
2226* C      THIS SUBROUTINE COMPUTES THE BESSELFUNC-00032070
2227* C      TION-PART OF THE INTEGRANDS FOR THE    00032080
2228* C      INTEGRALS COMPUTED IN INGRAL.          00032090
2229* C      FOR IL=1 THIS PART IS# JO(XR)*J1(X)   00032100
2230* C      FOR IL=2 THIS PART IS# J1(XR)*J1(X)   00032110
2231* C      COMPUTED RESULTS ARE DELIVERED AS FPIGR, 00032120
2232* C      EXP1 AND EXP2 IN COMMON/IGRAN/         00032130
2233* C      THE SUBROUTINE CALLS IN FUNCTION BESS. 00032140
2234* C-----00032150
2235* C      REAL LOAD,NU,ACCUR(3),K5                00032160
2236* C      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)00032170
2237* C      I,K5(10),E(10),AL(9),THICK(9),RADIUS(10) 00032180
2238* C      COMMON/IGRAN/TO,V0,SO,U0,T1,V1,S1,U1,TQ1,SQ1,FPIGR,EXP1,EXP2 00032190
2239* C      IF(LAYER.NE.1) GO TO 20                 00032200
2240* C      TO = TO-2.0*NU(1)                        00032210
2241* C      V0 = V0-1.0                             00032220
2242* C      T1 = T1-1.0+2.0*NU(1)                   00032230
2243* C      V1 = V1+1.0                             00032240
2244* C      TQ1 = TQ1-1.0                           00032250
2245* C      20 IF(R.LT.ACCUR(1)) GO TO 40             00032260
2246* C      IF(IL.EQ.2) GO TO 30                    00032270
2247* C      FPIGR = BESS(0,X*R)*BESS(1,X)/X          00032280
2248* C      GO TO 60                                00032290
2249* C      30 FPIGR = BESS(1,X*R)*BESS(1,X)/(X*R)    00032300
2250* C      GO TO 60                                00032310
2251* C      40 IF(IL.EQ.2) GO TO 50                  00032320
2252* C      FPIGR = BESS(1,X)/X                     00032330
2253* C      GO TO 60                                00032340
2254* C      50 FPIGR = 0.5*BESS(1,X)                 00032350
2255* C      60 IF(NLAYS.EQ.LAYER) GO TO 70           00032360
2256* C      IF(ABS(X*(2.0*H(LAYER)-Z)).GT.70.0)GO TO 70 00032370
2257* C      EXP1 = EXP(-X*(2.0*H(LAYER)-Z))          00032380
2258* C      IF((X*Z).GT.70.0)GO TO 90              00032390
2259* C      EXP2 = EXP(-X*Z)                        00032400
2260* C      GO TO 100                               00032410
2261* C      70 IF((X*Z).GT.70.0)GO TO 80            00032420
2262* C      EXP1 = 0.0                              00032430
2263* C      EXP2 = EXP(-X*Z)                        00032440
2264* C      GO TO 100                               00032450
2265* C      80 EXP1 = 0.0                           00032460
2266* C      90 EXP2 = 0.0                           00032470
2267* C      100 RETURN                             00032480
2268* C      END                                     00032490
2269* C      FUNCTION BESS(N,X)                      00032500
2270* C-----00032510
2271* C      THE BESSEL FUNCTIONS JO(X) AND J1(X) ARE 00032520
2272* C      EVALUATED FROM THEIR CHEBYSHEV SERIES.  00032530
2273* C      (SEE CLENSHAW,MATH. TABLES-VOL.5,     00032540
2274* C      CHEBYSHEV SERIES FOR MATH. FUNCTIONS    00032550
2275* C      NPL-DSIR).                              00032560
2276* C      THIS PROGRAM SELECTS THE APPROPRIATE   00032570
2277* C      CHEBYSHEV CONSTANTS ACCORDING TO WHETHER 00032580

```

```

2278* C                                N=0 OR N=1 AND WHETHER X IS GREATER OR 00032590
2279* C                                LESS THAN 8.0 AND CALLS IN FUNCTION CHEB 00032600
2280* C                                TO SUM THE SERIES. 00032610
2281* C----- 00032620
2282*      DOUBLE PRECISION B(12,2),BP(5,2),BQ(5,2),Z 00032630
2283*      DATA B /-.3D-8,.76D-7,-.1762D-5,.32460D-4,-.460626D-3,.4819180D-2, 00032640
2284*      1-.34893769D-1,.158067102D+0,-.370094994D+0,.265178613D+0, 00032650
2285*      2-.8723442D-2,.315455943D+0,-.1D-8,.29D-7,-.762D-6,.15887D-4, 00032660
2286*      3-.260444D-3,.3240270D-2,-.29175525D-1,.177709117D+0,-.661443934D+0 00032670
2287*      4,.1.28799410D+0,-1.19180116D+0,1.29671754D+0/ 00032680
2288*      DATA BP/.2D-8,-.52D-7,.3075D-5,-.536522D-3,1.99892070D+0, 00032690
2289*      1-.2D-8,.62D-7,-.3987D-5,.898990D-3,2.00180608D+0/ 00032700
2290*      DATA BQ/-.1D-8,.18D-7,-.741D-6,.68385D-4,-.31111709D-1, 00032710
2291*      1.1D-8,-.21D-7,.914D-6,-.96277D-4,.93555574D-1/ 00032720
2292*      M = N+1 00032730
2293*      IF(X-8.0) 1,1,2 00032740
2294*      1 Z = X*X*0.0625-2.0 00032750
2295*      BESS = CHEB(B(1,M),12,Z) 00032760
2296*      IF(N.EQ.1) BESS = 0.125*X*BESS 00032770
2297*      RETURN 00032780
2298*      2 Z = 256.0/(X*X)-2.0 00032790
2299*      X1 = X-0.78539816 00032800
2300*      IF(N.EQ.1) XI = XI-1.5707963 00032810
2301*      BESS = (0.79788456/SQRT(X))*(CHEB(BP(1,M),5,Z)*COS(XI)-8.0* 00032820
2302*      1 CHEB(BQ(1,M),5,Z)*SIN(XI)/X) 00032830
2303*      RETURN 00032840
2304*      END 00032850
2305*      FUNCTION CHEB(A,N,Z) 00032860
2306* C----- 00032870
2307* C                                THIS SUBPROGRAM EVALUATES THE CHEBYSHEV 00032880
2308* C                                SERIES USING THE RECURRENCE RELATION 00032890
2309* C                                TECHNIQUE (SEE CLENSHAW NPL MATH. TABLES 00032900
2310* C                                VOLUME 5 PAGE 9). 00032910
2311* C----- 00032920
2312*      DOUBLE PRECISION A(1),B(14),Z 00032930
2313*      B(1)=0.0D+0 00032940
2314*      B(2)=0.0D+0 00032950
2315*      DO 1 I=1,N 00032960
2316*      B(I+2)=Z*B(I+1)-B(I)+A(I) 00032970
2317*      1 CONTINUE 00032980
2318*      CHEB = 0.5D0*(B(N+2)-B(N)) 00032990
2319*      RETURN 00033000
2320*      END 00033010
2321*      REAL FUNCTION IGRAND (X,LABEL) 00033020
2322* C----- 00033030
2323* C                                THIS SUBROUTINE COMPUTES THE INTEGRANDS 00033040
2324* C                                FOR THE INTEGRALS COMPUTED IN INGRAL. 00033050
2325* C                                USE IS MADE OF THE RESULTS OF FPIGRA AND 00033060
2326* C                                MATRIX STORED IN COMMON/IGRAN/. 00033070
2327* C----- 00033080
2328*      REAL LOAD,NU,ACCUR(3),K5 00033090
2329*      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9) 00033100
2330*      1,K5(10),E(10),AL(9),THICK(9),RADIUS(10) 00033110

```

```

2331*      COMMON/IGRAN/T0,V0,S0,U0,T1,V1,S1,U1,TQ1,SQ1,FPIGR,EXP1,EXP2      00033120
2332*      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170)00033130
2333*      1,LABEL      00033140
2334*      10 IGRAND =FPIGR*X*((U0*(K5(LAYER)-X*Z)-S0)*EXP1+(T0+V0*(K5(LAYER)+X*00033150
2335*      1Z))*EXP2)      00033160
2336*      RETURN      00033170
2337*      20 IGRAND =FPIGR*X*(U0*EXP1+V0*EXP2)      00033180
2338*      RETURN      00033190
2339*      30 IGRAND =FPIGR*((U0*(2.0*K5(LAYER)-X*Z)-S0)*EXP1-(T0+V0*(2.0*K5(LAY00033200
2340*      1ER)+X*Z))*EXP2)      00033210
2341*      RETURN      00033220
2342*      40 IGRAND =FPIGR*((S0+U0*(1.0+X*Z))*EXP1+(V0*(1.0-X*Z)-T0)*EXP2)      00033230
2343*      RETURN      00033240
2344*      50 IGRAND =FPIGR*X*R*((S0+U0*(2.*NU(LAYER)+X*Z))*EXP1+(T0+V0*(X*Z-2.*00033250
2345*      1NU(LAYER)))*EXP2)      00033260
2346*      RETURN      00033270
2347*      60 IGRAND =FPIGR*X*((S1+U1*(2.0*NU(LAYER)+X*Z))*EXP1+(T1+V1*(X*Z-2.0*00033280
2348*      1NU(LAYER)))*EXP2)      00033290
2349*      RETURN      00033300
2350*      70 IGRAND =FPIGR*((S1+U1*(1.0+X*Z))*EXP1+(V1*(1.0-X*Z)-T1)*EXP2)      00033310
2351*      RETURN      00033320
2352*      80 IGRAND =FPIGR*X*R*((S1+U1*(1.0+X*Z))*EXP1+(V1*(1.0-X*Z)-T1)*EXP2)      00033330
2353*      RETURN      00033340
2354*      90 IGRAND =FPIGR*X*R*(U1*EXP1+V1*EXP2)      00033350
2355*      RETURN      00033360
2356*      100 IGRAND =FPIGR*((S1+U1*(2.*NU(LAYER)+X*Z))*EXP1+(T1+V1*(X*Z-2.*NU(L00033370
2357*      1AYER)))*EXP2)      00033380
2358*      RETURN      00033390
2359*      110 IGRAND =FPIGR*(U1*EXP1-V1*EXP2)      00033400
2360*      RETURN      00033410
2361*      120 IGRAND =FPIGR*((S1+U1*(1.+X*Z))*EXP1+(V1*(1.-X*Z)-T1)*EXP2)/X      00033420
2362*      RETURN      00033430
2363*      130 IGRAND =FPIGR*X*(SQ1*EXP1-TQ1*EXP2)      00033440
2364*      RETURN      00033450
2365*      140 IGRAND =FPIGR*(SQ1*EXP1+TQ1*EXP2)      00033460
2366*      RETURN      00033470
2367*      150 IGRAND =FPIGR*X*R*(SQ1*EXP1+TQ1*EXP2)      00033480
2368*      RETURN      00033490
2369*      160 IGRAND =FPIGR*(SQ1*EXP1-TQ1*EXP2)      00033500
2370*      RETURN      00033510
2371*      170 IGRAND =FPIGR*(SQ1*EXP1+TQ1*EXP2)/X      00033520
2372*      RETURN      00033530
2373*      END      00033540
2374*      SUBROUTINE CALC(INT,V,R,MU,RADI,FT,LOAD,HOSTRS,PSIO,Z)      00033550
2375*      C-----00033560
2376*      C      COMPONENTS OF THE STRESSES,STRAINS AND      00033570
2377*      C      THIS SUBROUTINE COMPUTES THE CYLINDRICAL      00033580
2378*      C      DISPLACEMENTS FROM THE 17 INTEGRALS STO-      00033590
2379*      C      RED IN INT. THESE CALCULATED COMPONENTS      00033600
2380*      C      ARE STORED IN V AND OUTPUTTED.      00033610
2381*      C-----00033620
2382*      REAL INT(17),V(15),MU,LOAD,C(6)      00033630
2383*      INTEGER FM(19),FMT(5),T(12)      00033640

```

```

2384* LOGICAL STRESS,EPS,RLOW 00033650
2385* COMMON/STRTDA/STRESS(27),EPS(17),RLOW,ST,CT,L,ACC 00033660
2386* COMMON/TAPE/NOTOUT 00033670
2387* DATA FM(1),FMT,FM(19),T/ 00033680
2388* +' (IX ',',E12',',.4,1','0X ',',12X',' ',1','X) ', 00033690
2389* +'DISP','LACE','MENT','S ', 00033700
2390* +' S','TRES','SES ',',', 00033710
2391* +' S','TRAI','NS ',',', 00033720
2392* DO 10 I=1,15 00033730
2393* 10 V(I)=0.0 00033740
2394* IF((STRESS( 4).OR.STRESS( 5).OR.STRESS( 7).OR.STRESS(10).OR. 00033750
2395* + STRESS(11)).AND.(.NOT.RLOW)) FCT=(2.0*INT(12)-INT( 7)-2.0*INT( 00033760
2396* + 14)+4.0*INT(17))/R 00033770
2397* IF(.NOT.STRESS( 1)) GO TO 20 00033780
2398* V( 1)=FT*RADI*CT*(2.0*INT(17)+INT(12)-INT( 7)) 00033790
2399* IF(RLOW) GO TO 20 00033800
2400* V( 1)=V( 1)-FT*R*RADI*INT( 4) 00033810
2401* 20 IF(STRESS( 2)) V( 2)=FT*RADI*ST*(2.0*(INT(17)-INT(14))+INT(12)) 00033820
2402* IF(.NOT.STRESS( 3)) GO TO 30 00033830
2403* V( 3)=-FT*RADI*INT( 3) 00033840
2404* IF(RLOW) GO TO 30 00033850
2405* V( 3)=V( 3)+FT*R*RADI*CT*((2.0-2.0*MU)*INT(11)-INT(10)) 00033860
2406* 30 IF(.NOT.STRESS( 4)) GO TO 40 00033870
2407* V( 4)=CT*(INT( 8)+2.0*MU*INT( 9))+INT( 1)+INT( 4)-2.0*INT( 2) 00033880
2408* IF(RLOW) GO TO 40 00033890
2409* V( 4)=V( 4)-CT*FCT 00033900
2410* 40 IF(.NOT.STRESS( 5)) GO TO 50 00033910
2411* W( 5)=CT*2.0*MU*INT( 9)-2.0*MU*INT( 2)-INT( 4) 00033920
2412* IF(RLOW) GO TO 50 00033930
2413* V( 5)=V( 5)+CT*FCT 00033940
2414* 50 IF(.NOT.STRESS( 7)) GO TO 60 00033950
2415* V( 7)=ST*INT(15) 00033960
2416* IF(RLOW) GO TO 60 00033970
2417* V( 7)=V( 7)-ST*FCT 00033980
2418* 60 IF(.NOT.STRESS(10)) GO TO 70 00033990
2419* V(10)=FT*(CT*INT( 8)+INT( 1)+INT( 4)-(2.0-2.0*MU)*INT( 2)) 00034000
2420* IF(RLOW) GO TO 70 00034010
2421* V(10)=V(10)-FT*CT*FCT 00034020
2422* 70 IF(.NOT.STRESS(11)) GO TO 80 00034030
2423* V(11)=-FT*INT( 4) 00034040
2424* IF(RLOW) GO TO 80 00034050
2425* V(11)=V(11)+FT*CT*FCT 00034060
2426* 80 IF(STRESS(12)) V(12)=FT*(CT*((2.0-4.0*MU)*INT( 9)-INT( 8))+ 00034070
2427* + 2.0*MU*INT( 2)-INT( 1)) 00034080
2428* IF(Z.LT.ACC) GO TO 90 00034090
2429* IF(STRESS( 6)) V( 6)=CT*((2.0-2.0*MU)*INT( 9)-INT( 8))-INT( 1) 00034100
2430* IF(STRESS( 8)) V( 8)=CT*(INT(16)+INT(10)-INT( 6))-INT( 5) 00034110
2431* IF(STRESS( 9)) V( 9)=ST*INT(16)-INT(13)+INT(10)) 00034120
2432* GO TO 110 00034130
2433* 90 IF(ABS(R-1.0).LT.ACC) GO TO 100 00034140
2434* IF(R.GT.1.0) GO TO 110 00034150
2435* IF(STRESS( 6)) V( 6)=-LOAD 00034160
2436* IF(STRESS( 8)) V( 8)=-HOSTRS*COS(PSI0) 00034170

```

2437*	IF(STRESS(9)) V(9)=HOSTRS*SIN(PSI0)	00034180
2438*	GO TO 110	00034190
2439*	100 IF(STRESS(6)) V(6)=-0.5*LOAD	00034200
2440*	IF(STRESS(8)) V(8)=-0.5*HOSTRS*COS(PSI0)	00034210
2441*	IF(STRESS(9)) V(9)= 0.5*HOSTRS*SIN(PSI0)	00034220
2442*	110 IF(STRESS(13)) V(13)=FT*V(7)	00034230
2443*	IF(STRESS(14)) V(14)=FT*V(8)	00034240
2444*	IF(STRESS(15)) V(15)=FT*V(9)	00034250
2445*	DO 120 I=2,18	00034260
2446*	120 FM(I)=T(8)	00034270
2447*	DO 130 I=4,16,3	00034280
2448*	130 FM(I)=FMT(3)	00034290
2449*	K=0	00034300
2450*	J=0	00034310
2451*	DO 210 I=1,15	00034320
2452*	J=J+1	00034330
2453*	IF(I-4) 190,160,140	00034340
2454*	140 IF(I-10) 190,150,190	00034350
2455*	150 IF(K.EQ.0) GO TO 180	00034360
2456*	WRITE(NOUT,9010) (T(J),J=5,8)	00034370
2457*	WRITE(NOUT,FM) (C(J),J=1,K)	00034380
2458*	GO TO 170	00034390
2459*	160 IF(K.EQ.0) GO TO 180	00034400
2460*	WRITE(NOUT,9000) (T(J),J=1,4)	00034410
2461*	WRITE(NOUT,9013) (C(J),J=1,K)	
2462*	170 K=0	00034430
2463*	180 J=1	00034440
2464*	190 M=3*J	00034450
2465*	IF(.NOT.STRESS(I)) GO TO 200	00034460
2466*	K=K+1	00034470
2467*	C(K)=V(I)	00034480
2468*	FM(M-1)=FMT(1)	00034490
2469*	FM(M)=FMT(2)	00034500
2470*	GO TO 210	00034510
2471*	200 FM(M-1)=FMT(4)	00034520
2472*	FM(M)=FMT(5)	00034530
2473*	210 CONTINUE	00034540
2474*	IF(K.EQ.0) RETURN	00034550
2475*	WRITE(NOUT,9010) (T(J),J=9,12)	00034560
2476*	WRITE(NOUT,FM) (C(J),J=1,K)	00034570
2477*	RETURN	00034580
2478*	9000 FORMAT(1X,4A4/5X,'RADIAL',12X,'TANGENTIAL',14X,'VERTICAL')	00034590
2479*	9010 FORMAT(1X,4A4/5X,'RADIAL',12X,'TANGENTIAL',14X,'VERTICAL',12X,'RAD00034600	
2480*	+/TANG.',11X,'RAD./VERT.',12X,'TANG./VERT.')	00034610
2481*	9013 FORMAT(1X,3(E12.4,10X))	
2482*	END	00034620
2483*	SUBROUTINE OUTPUT(EPS,C,K,L)	00034630
2484*	C-----	00034640
2485*	C THIS SUBROUTINE OUTPUTS BY MEANS OF THREE	00034650
2486*	C SUBSEQUENT CALLS FROM THE MAIN PROGRAM	00034660
2487*	C THE TOTAL STRESSES,STRAINS AND DISPLACE-	00034670
2488*	C MENTS.	00034680
2489*	C-----	00034690

2490*	INTEGER FM(16),FMT(8)	00034700
2491*	LOGICAL EPS(6)	00034710
2492*	DIMENSION C(6),TKST(6,4)	00034720
2493*	COMMON/TAPE/NOU	00034730
2494*	DATA TKST/	00034740
2495*	1' T O', ' T A', ' L ', ' S T ', ' R E ', ' S S ',	00034750
2496*	2' T O', ' T A', ' L ', ' S T ', ' R A ', ' I N ',	00034760
2497*	3' T O', ' T A', ' L ', ' D I ', ' S P ', ' L A ',	00034770
2498*	4' C E ', ' M E ', ' N T ', ' ', ' ', ' ', ' ',	00034780
2499*	DATA FMT,FM(16)/	00034790
2500*	1' (6A4', '12X', ' ', ' ', 'E12', '1.3 ', ' (12A', '4,48', 'X ', ' ') '/	00034800
2501*	IF(L.NE.3) GO TO 10	00034810
2502*	FM(1)=FMT(6)	00034820
2503*	FM(2)=FMT(7)	00034830
2504*	FM(3)=FMT(8)	00034840
2505*	GO TO 20	00034850
2506*	10 FM(1)=FMT(1)	00034860
2507*	FM(2)=FMT(3)	00034870
2508*	FM(3)=FMT(3)	00034880
2509*	20 N=0	00034890
2510*	M=2*K+2	00034900
2511*	DO 40 I=4,M,2	00034910
2512*	J=I/2-1	00034920
2513*	IF(.NOT.EPS(J)) GO TO 30	00034930
2514*	FM(I)=FMT(4)	00034940
2515*	FM(I+1)=FMT(5)	00034950
2516*	N=N+1	00034960
2517*	C(N)=C(J)	00034970
2518*	GO TO 40	00034980
2519*	30 FM(I)=FMT(2)	00034990
2520*	FM(I+1)=FMT(3)	00035000
2521*	40 CONTINUE	00035010
2522*	IF(L.EQ.3) GO TO 60	00035020
2523*	IF(N.EQ.0) GO TO 50	00035030
2524*	WRITE(NOUT,FM) (TKST(I,L),I=1,6),(C(I),I=1,N)	00035040
2525*	RETURN	00035050
2526*	50 WRITE(NOUT,FM) (TKST(I,L),I=1,6)	00035060
2527*	RETURN	00035070
2528*	60 DO 70 I=10,15	00035080
2529*	70 FM(I)=FMT(3)	00035090
2530*	IF(N.EQ.0) GO TO 80	00035100
2531*	WRITE(NOUT,FM) (TKST(I,3),I=1,6),(TKST(I,4),I=1,6),(C(I),I=1,N)	00035110
2532*	RETURN	00035120
2533*	80 WRITE(NOUT,FM) (TKST(I,3),I=1,6),(TKST(I,4),I=1,6)	00035130
2534*	RETURN	00035140
2535*	END	00035150
2536*	SUBROUTINE JACOBI (H,U,ND,N,IVC,W,IQ)	00035160
2537*	C-----	00035170
2538*	C SUBROUTINE JACOBI TO COMPUTE EIGENVALUES	00035180
2539*	C AND EIGENVECTORS OF A SYMMETRIC MATRIX.	00035190
2540*	C H IS THE GIVEN MATRIX,THE DIAGONAL OF	00035200
2541*	C WHICH CONTAINS AFTER THE ITERATION THE	00035210
2542*	C EIGENVALUES OF H.	00035220

2543* C	U IS THE MATRIX,THE COLUMNS OF WHICH ARE	00035230
2544* C	THE EIGENVECTORS OF H.	00035240
2545* C	N AND ND ARE THE DIMENSIONS OF THE ACTUAL	00035250
2546* C	MATRIX AND THE ONE USED IN THE DIMENSION-	00035260
2547* C	STATEMENT OF THE CALLINGPROGRAM	00035270
2548* C	RESPECTIVELY.	00035280
2549* C	IVEC=0 IF NO EIGENVECTORS ARE REQUIRED,	00035290
2550* C	IVEC=1 IF THE EIGENVECTORS SHOULD BE	00035300
2551* C	CALCULATED.	00035310
2552* C	THE ACCURACY OF THE EIGENVALUES IS ABOUT	00035320
2553* C	1.0E-6,THE ACCURACY OF AN EIGENVECTOR IS	00035330
2554* C	ABOUT 1.0E-6/D,WHERE D IS THE MINIMUM-	00035340
2555* C	DISTANCE OF THE CORRESPONDING EIGENVALUE	00035350
2556* C	FROM THE OTHER EIGENVALUES.	00035360
2557* C	W AND IQ ARE WORKINGSPACES,WHICH SHOULD BE	00035370
2558* C	DIMENSIONED IN THE CALLING PROGRAM.	00035380
2559* C	-----	00035390
2560*	REAL H(ND,ND),U(ND,ND),W(ND)	00035400
2561*	INTEGER IQ(ND)	00035410
2562*	DOUBLE PRECISION TA,SI,CO,Z,Y,HTE,UTE	00035420
2563*	AN =N	00035430
2564*	NM11=N-1	00035440
2565*	IF(IVEC-1) 60,10,60	00035450
2566*	10 DO 50 I=1,N	00035460
2567*	DO 40 J=1,N	00035470
2568*	IF(I-J) 30,20,30	00035480
2569*	20 U(I,J)=1.0	00035490
2570*	GO TO 40	00035500
2571*	30 U(I,J)=0.0	00035510
2572*	40 CONTINUE	00035520
2573*	50 CONTINUE	00035530
2574*	60 DO 90 I=1,NM11	00035540
2575*	W(I)=0.0	00035550
2576*	IPL1=I+1	00035560
2577*	DO 80 J=IPL1,N	00035570
2578*	IF(W(I)-ABS(H(I,J))) 70,70,80	00035580
2579*	70 W(I)=ABS(H(I,J))	00035590
2580*	IQ(I)=J	00035600
2581*	80 CONTINUE	00035610
2582*	90 CONTINUE	00035620
2583*	100 DO 120 I=1,NM11	00035630
2584*	IF(I.EQ.1) GO TO 110	00035640
2585*	IF(XMAX.GE.W(I)) GO TO 120	00035650
2586*	110 XMAX=W(I)	00035660
2587*	IPIV=I	00035670
2588*	JPIV=IQ(I)	00035680
2589*	120 CONTINUE	00035690
2590*	IF(XMAX-1.E-12/AN) 170,170,130	00035700
2591*	130 Z =H(IPIV,IPIV)-H(JPIV,JPIV)	00035710
2592*	Y = 2.000*DBLE(H(IPIV,JPIV))	00035720
2593*	TA =Y/(DABS(Z)+DSQRT(Z*Z+Y*Y))	00035730
2594*	IF(Z.LT.0.000) TA=-TA	00035740
2595*	CO =1.00/DSQRT(1.00+TA*TA)	00035750

2596*	SI =TA*CO	00035760
2597*	HII=H(IPIV,IPIV)	00035770
2598*	HJJ=H(JPIV,JPIV)	00035780
2599*	HIJ=H(IPIV,JPIV)	00035790
2600*	DO 140 K=1,N	00035800
2601*	HTE=H(K,IPIV)	00035810
2602*	H(K,IPIV)=DBLE(H(K,IPIV))*CO+DBLE(H(K,JPIV))*SI	00035820
2603*	H(K,JPIV)=DBLE(H(K,JPIV))*CO-HTE*SI	00035830
2604*	H(IPIV,K)=H(K,IPIV)	00035840
2605*	H(JPIV,K)=H(K,JPIV)	00035850
2606*	140 CONTINUE	00035860
2607*	H(IPIV,JPIV)=0.0	00035870
2608*	H(JPIV,IPIV)=0.0	00035880
2609*	AA=DBLE(HIJ)*TA	00035890
2610*	H(IPIV,IPIV)=HII+AA	00035900
2611*	H(JPIV,JPIV)=HJJ-AA	00035910
2612*	IF(IVEC) 60,60,150	00035920
2613*	150 DO 160 K=1,N	00035930
2614*	UTE=U(K,IPIV)	00035940
2615*	U(K,IPIV)=DBLE(U(K,IPIV))*CO+DBLE(U(K,JPIV))*SI	00035950
2616*	U(K,JPIV)=DBLE(U(K,JPIV))*CO-UTE*SI	00035960
2617*	160 CONTINUE	00035970
2618*	GO TO 60	00035980
2619*	170 RETURN	00035990
2620*	END	00036000
2621*	SUBROUTINE ESORT (H,U,ND,N,IVEC,W,IQ)	00036010
2622*	C-----	00036020
2623*	C SUBROUTINE ESORT.	00036030
2624*	C THIS ROUTINE SORTS EIGENVALUES (AND EIGEN	00036040
2625*	C VECTORS) OBTAINED FROM SUBROUTINE JACOBI.	00036050
2626*	C H = ORIGINAL MATRIX(ND,ND).	00036060
2627*	C U = EIGENVECTORMATRIX(ND,ND).	00036070
2628*	C ND = MAX. DIMENSION OF MATRICES.	00036080
2629*	C N = ACTUAL DIMENSION OF MATRICES.	00036090
2630*	C IVEC=1 WITH EIGENVECTORS,	00036100
2631*	C =0 NO EIGENVECTORS.	00036110
2632*	C W = WORKINGSPACE(ND).	00036120
2633*	C IQ = WORKINGSPACE(ND).	00036130
2634*	C-----	00036140
2635*	REAL H(ND,ND),U(ND,ND),W(ND),DUMMY	00036150
2636*	INTEGER N,IQ(ND),FDUMMY,I,J,K,IVEC	00036160
2637*	LOGICAL LOGIC	00036170
2638*	C	00036180
2639*	DO 10 I=1,N	00036190
2640*	W(I)=H(I,I)	00036200
2641*	10 IQ(I)=I	00036210
2642*	J=N	00036220
2643*	20 LOGIC=.FALSE.	00036230
2644*	K=J	00036240
2645*	DO 30 I=2,K	00036250
2646*	IF(W(I-1).GE.W(I)) GO TO 30	00036260
2647*	LOGIC=.TRUE.	00036270
2648*	DUMMY=W(I-1)	00036280

2649*	W(I-1)=W(I)	00036290
2650*	W(I)=DUMMY	00036300
2651*	FDUMMY=IQ(I-1)	00036310
2652*	IQ(I-1)=IQ(I)	00036320
2653*	IQ(I)=FDUMMY	00036330
2654*	J=I-1	00036340
2655*	30 CONTINUE	00036350
2656*	IF (LOGIC) GO TO 20	00036360
2657*	IF (IVEC.EQ.0) GO TO 60	00036370
2658*	DO 40 I=1,N	00036380
2659*	K=IQ(I)	00036390
2660*	DO 40 J=1,N	00036400
2661*	40 H(J,I)=U(J,K)	00036410
2662*	DO 50 I=1,N	00036420
2663*	DO 50 J=1,N	00036430
2664*	U(I,J)=H(I,J)	00036440
2665*	50 H(I,J)=0.0	00036450
2666*	60 DO 70 I=1,N	00036460
2667*	70 H(I,I)=W(I)	00036470
2668*	RETURN	00036480
2669*	END	00036490
2670*	SUBROUTINE RNN(AA,BB,RN,ALOAD,ALIN,CAREA)	00036500
2671*	C FLEXIBLE PAVEMENT ROUTINE	00036510
2672*	C COMPUTES THE LIMITING VERTICAL STRAIN AS A FUNCTION OF LOAD	00036520
2673*	C REPETITION, INDEPENDENT OF THE SUBGRADE YOUNG'S MODULUS.	00036530
2674*	C COMMON VSTR,STRL,ITER,STRL2	00036540
2675*	C COMMON/RADIAL/STSL,DSM,FS,SWL	00036550
2676*	WRITE(6,10)	00036560
2677*	10 FORMAT(// 'PROGRAM USING STRAIN REPETITION NUMBER '//)	00036570
2678*	CALL FFRD(5,AA,1)	
2679*	CALL FFRD(5,BB,2)	
2680*	CALL FFRD(5,RN,2)	
2681*	CALL FFRD(5,ALOAD,2)	
2682*	CALL FFRD(5,ALIN,2)	
2683*	CALL FFRD(5,CAREA,2)	
2684*	CALL FFRD(5,DSW,2)	
2685*	CALL FFRD(5,SWL,2)	
2686*	YY=ALOG10(RN)	00036600
2687*	Q=AA*YY+BB	00036610
2688*	STRL=10.**Q	00036620
2689*	RETURN	00036630
2690*	END	00036640
2691*	SUBROUTINE RPAL(ALOAD,ALIN,CAREA)	00036650
2692*	C RIGID PAVEMENT ALLOWABLE LOAD ROUTINE.	00036660
2693*	C COMPUTES THE LIMITING RADIAL STRAIN.	00036670
2694*	C COMMON/RADIAL/STSL,DSM,FS,SWL	00036680
2695*	DATA A,B/0.58901,0.35486/	00036690
2696*	CALL FFRD(5,DSM,1)	
2697*	CALL FFRD(5,FAC,2)	
2698*	CALL FFRD(5,YRN,2)	
2699*	CALL FFRD(5,FS,2)	
2700*	CALL FFRD(5,ALOAD,2)	
2701*	CALL FFRD(5,ALIN,2)	

```

2702*      CALL FFRD(5,CAREA,2)
2703*      CALL FFRD(5,SWL,2)
2704*      RN=20.*YRN
2705*      COV=RN/FAC
2706*      STSL=FS/(A+B*ALOG10(COV))
2707*      WRITE(6,2)A,B,YRN,FAC,COV
2708*  2    FORMAT(///60(1H*)//,20X,'A' = ',F10.6//,
2709*      1    20X,'B' = ',F10.6//,
2710*      2    20X,'YRN' = ',F10.0//,
2711*      3    20X,'FAC' = ',F10.4//,
2712*      4    20X,'COV' = ',F10.0//60(1H*)////)
2713*      RETURN
2714*      END
2715*      SUBROUTINE FLEX(ES,EA,ALOAD,ALIN,CAREA,XS,AS,BS,YRN,PRATIO)
2716*  C      FLEXIBLE PAVEMENT ROUTINE
2717*      DIMENSION AO(8),BO(8)
2718*      COMMON VSTR,STRL,ITER,STRL2
2719*      COMMON /RADIAL/STSL,DSM,FS,SWL
2720*      DATA AO/2.32E-03,-9.6347304E-05,2.9507649E-05,-4.2586473E-06,
2721*      13.0210971E-07,-1.1453280E-08,2.2311010E-10,-1.7523733E-12/
2722*      DATA BO/1.54E-01,-1.7032892E-02,2.8018403E-03,-3.5283369E-04,
2723*      12.5877567E-05,-1.0485427E-06,2.1756275E-08,-1.7990976E-10/
2724*      CALL FFRD(5,ES,1)
2725*      CALL FFRD(5,EA,2)
2726*      CALL FFRD(5,YRN,2)
2727*      CALL FFRD(5,ALOAD,2)
2728*      CALL FFRD(5,ALIN,2)
2729*      CALL FFRD(5,CAREA,2)
2730*      CALL FFRD(5,DSM,2)
2731*      CALL FFRD(5,SWL,2)
2732*      CALL FFRD(5,PRATIO,2)
2733*      WRITE(6,11)YRN
2734*  11    FORMAT(//,'PROGRAM USING ',F10.0,3X,'REPETITIONS.....',//)
2735*      XS=ES/1000.
2736*      AS=AO(1)+AO(2)*XS+AO(3)*XS**2+AO(4)*XS**3+AO(5)*XS**4+
2737*      1AO(6)*XS**5+AO(7)*XS**6+AO(8)*XS**7
2738*      BS=BO(1)+BO(2)*XS+BO(3)*XS**2+BO(4)*XS**3+BO(5)*XS**4+BO(6)
2739*      1XS**5+BO(7)*XS**6+BO(8)*XS**7
2740*      BSNEG=-BS
2741*      STRL=AS*YRN**BSNEG
2742*      ACREPS=ALOG10(YRN*20./PRATIO)
2743*      EAL=EA/14.22
2744*      EALOG=ALOG10(EAL)
2745*      STRL2=(ACREPS+2.665*EALOG+0.392)/5.
2746*      STRL2=-STRL2
2747*      STRL2=10.**STRL2
2748*      RETURN
2749*      END
2750*      SUBROUTINE POST(XTEMP,YTEMP,NPOS2,LAY,AXX,AYY,DEP,ETAA)
2751*  C      ROUTINE THAT GENERATES THE COMPUTATIONAL POSITIONS ONCE THE
2752*  C      FINAL LOAD OR THICKNESS IS FOUND.
2753*      REAL NU,ACCUR(3),LOAD,K5
2754*      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)

```

2755*	+	K5(10),E(10),AL(9),THICK(9),RADIUS(10)	00037160
2756*		DIMENSION XTEMP(10),YTEMP(10),LAY(100),AXX(100),AYY(100),	00037170
2757*	1	DEP(100),ETAA(100)	00037180
2758*		NLOC=NPOS2	00037190
2759*		NPOS2=NLAYS*2-1	00037200
2760*		M1=2	00037210
2761*		M2=NPOS2	00037220
2762*		KPOS=0	00037230
2763*		K=0	00037240
2764*		WRITE(6,13)THICK	00037250
2765*	13	FORMAT(// 'THICK ARRAY = ',/9F10.2//)	00037260
2766*		DO 48 I=1,NLOC	00037270
2767*		LAY(K+1)=1	00037280
2768*		AXX(K+1)=XTEMP(I)	00037290
2769*		AYY(K+1)=YTEMP(I)	00037300
2770*		DEP(K+1)=0.	00037310
2771*		ETAA(K+1)=0.	00037320
2772*		KK=1	00037330
2773*		TSUM=0.	00037340
2774*		DO 49 J=M1,M2,2	00037350
2775*		LAY(J)=(J-KPOS)/2	00037360
2776*		LAY(J+1)=(J-KPOS)/2+1	00037370
2777*		AXX(J)=XTEMP(I)	00037380
2778*		AXX(J+1)=XTEMP(I)	00037390
2779*		AYY(J)=YTEMP(I)	00037400
2780*		AYY(J+1)=YTEMP(I)	00037410
2781*		WRITE(6,12)KK,THICK(KK)	00037420
2782*	12	FORMAT('KK = ',15,5X,'THICK = ',F8.2//)	00037430
2783*		DEP(J)=THICK(KK)+TSUM	00037440
2784*		DEP(J+1)=THICK(KK)+TSUM	00037450
2785*		ETAA(J)=0.	00037460
2786*		ETAA(J+1)=0.	00037470
2787*		KK=KK+1	00037480
2788*		TSUM=DEP(J)	00037490
2789*	49	CONTINUE	00037500
2790*		IF(I.EQ.NLOC)GO TO 48	00037510
2791*		K=NPOS2+K	00037520
2792*		M1=M2+2	00037530
2793*		M2=NPOS2+M2	00037540
2794*		KPOS=KPOS+NPOS2	00037550
2795*	48	CONTINUE	00037560
2796*		NPOS2=M2	00037570
2797*		WRITE(6,10)	00037580
2798*		WRITE(6,11)(LAY(I),AXX(I),AYY(I),DEP(I),ETAA(I),I=1,M2)	00037590
2799*	10	FORMAT('SUBROUTINE POST-----')	00037600
2800*	11	FORMAT((110,4F15.2//)	00037610
2801*		RETURN	00037620
2802*		END	00037630
2803*		SUBROUTINE NFRD(IFC,NV,ID)	
2804*		COMMON/JRJ/ICRD(80),IFMT(10),IVAL(10),NL	
2805*		DATA IB/1H/,IC/1H.,	
2806*		IF(ID.NE.1) GO TO 10	
2807*		READ(IFC,2) (ICRD(I),I=1,80)	

```

2808*      2 FORMAT(80A1)
2809*      NL=1
2810*      10 IF(NL.GE.80) NV=0
2811*      IF(NL.GE.80) RETURN
2812* C
2813* C      FIND FIRST NONE BLANK CHARACTER
2814* C
2815*      DO 3 J=NL,80
2816*      IF(ICRD(J).NE.IB) GO TO 5
2817*      I=J
2818*      3 CONTINUE
2819*      NL=I+1
2820*      GO TO 10
2821*      5 NL=J
2822*      IF(ICRD(NL).NE.IC) GO TO 4
2823*      NV=0
2824*      NL=NL+1
2825*      RETURN
2826*      4 DO 6 J=NL,80
2827*      IF(ICRD(J).EQ.IB) GO TO 7
2828*      IF(ICRD(J).EQ.IC) GO TO 7
2829*      6 CONTINUE
2830*      J=81
2831*      7 K=NL
2832*      M=J-1
2833*      ENCODE(40,2,IVAL(1)) (ICRD(I),I=K,M)
2834*      NL=J+1
2835*      L=M-K+1
2836*      IF(L.LE.9) ENCODE(40,8,IFMT(1)) L
2837*      IF(L.GE.10) ENCODE(40,9,IFMT(1)) L
2838*      8 FORMAT(' (I',I1,') ')
2839*      9 FORMAT(' (I',I2,') ')
2840*      DECODE(L,IFMT,IVAL(1)) NV
2841*      RETURN
2842*      END
2843*      SUBROUTINE FFRD(IFC,F,ID)
2844*      COMMON/JRJ/ICRD(80),IFMT(10),IVAL(10),NL
2845*      DATA IB/1H /,IC/1H,/
2846*      IF(ID.NE.1) GO TO 10
2847*      READ(IFC,2) (ICRD(I),I=1,80)
2848*      2 FORMAT(80A1)
2849*      NL=1
2850*      10 IF(NL.GE.80) F=0.0
2851*      IF(NL.GE.80) RETURN
2852* C
2853* C      FIND FIRST NONE BLANK CHARACTER
2854* C
2855*      DO 3 J=NL,80
2856*      IF(ICRD(J).NE.IB) GO TO 5
2857*      I=J
2858*      3 CONTINUE
2859*      NL=I+1
2860*      GO TO 10

```

```

2861*      5 NL=J
2862*      IF(ICRD(NL).NE.IC) GO TO 4
2863*      F=0.0
2864*      NL=NL+1
2865*      RETURN
2866*      4 DO 6 J=NL,80
2867*      IF(ICRD(J).EQ.IB) GO TO 7
2868*      IF(ICRD(J).EQ.IC) GO TO 7
2869*      6 CONTINUE
2870*      J=81
2871*      7 K=NL
2872*      M=J-1
2873*      ENCODE(40,2,IVAL(1)) (ICRD(1),I=K,M)
2874*      NL=J+1
2875*      L=M-K+1
2876*      IF(L.LE.9) ENCODE(40,8,IFMT(1)) L
2877*      IF(L.GE.10) ENCODE(40,9,IFMT(1)) L
2878*      8 FORMAT('F',I1,'.0')
2879*      9 FORMAT('F',I2,'.0')
2880*      DECODE(L,IFMT,IVAL(1)) F
2881*      RETURN
2882*      END
2883*      BLOCK DATA
2884*      C-----00037640
2885*      C-----00037650
2886*      C      IN THE BLOCK DATA ARE STORED SUBSEQUENTLY#00037660
2887*      C      -THE ABSCISSAE FOR THE LEGENDRE-GAUSS 00037670
2888*      C      QUADRATURE,STARTING IN A WITH THE 2-ND 00037680
2889*      C      ORDER AND ENDING IN N WITH THE 15-TH 00037690
2890*      C      ORDER. 00037700
2891*      C      -THE ABSCISSAE FOR THE JACOBI-GAUSS 00037710
2892*      C      QUADRATURE OF THE 8-TH ORDER IN O. 00037720
2893*      C      -THE WEIGHTS FOR THE LEGENDRE-GAUSS 00037730
2894*      C      QUADRATURE,STARTING IN P WITH THE 2-ND 00037740
2895*      C      ORDER AND ENDING IN CC WITH THE 15-TH 00037750
2896*      C      ORDER. 00037760
2897*      C      -THE WEIGHTS FOR THE JACOBI-GAUSS QUADRA- 00037770
2898*      C      TURE OF THE 8-TH ORDER IN DD. 00037780
2899*      C      -THE FIRST 149 ZEROS OF JO IN EE AND FF 00037790
2900*      C      -THE FIRST 149 ZEROS OF JI IN GG AND HH 00037800
2901*      C-----00037810
2902*      REAL I,J,K,L,M,N 00037820
2903*      DIMENSION A(2),B(3),C(4),D(5),E(6),F(7),G(8),H(9),I(10),J(11),K(12) 00037830
2904*      L(13),M(14),N(15),O( 8),P(2),Q(3),R(4),S(5),T(6),U(7),V(8),W(9), 00037840
2905*      2X(10),Y(11),Z(12),AA(13),BB(14),CC(15),DD( 8),EE(119),FF(30),GG(11 00037850
2906*      39),HH(30) 00037860
2907*      COMMON/GAUSS/AGAUSS(16,16),HGAUSS(16,16) 00037870
2908*      COMMON/GEDATA/BZEROS(149,2),ZEROS(298) 00037880
2909*      EQUIVALENCE (AGAUSS(1, 2), A(1)),(AGAUSS(1, 3), B(1)), 00037890
2910*      1(AGAUSS(1, 4), C(1)),(AGAUSS(1, 5), D(1)),(AGAUSS(1, 6), E(1)), 00037900
2911*      2(AGAUSS(1, 7), F(1)),(AGAUSS(1, 8), G(1)),(AGAUSS(1, 9), H(1)), 00037910
2912*      3(AGAUSS(1,10), I(1)),(AGAUSS(1,11), J(1)),(AGAUSS(1,12), K(1)), 00037920
2913*      4(AGAUSS(1,13), L(1)),(AGAUSS(1,14), M(1)),(AGAUSS(1,15), N(1)), 00037930
2914*      5(AGAUSS(1,16), O(1)),(HGAUSS(1, 2), P(1)),(HGAUSS(1, 3), Q(1)), 00037940

```

```

2914* 6(HGAUSS(1, 4), R(1)),(HGAUSS(1, 5), S(1)),(HGAUSS(1, 6), T(1)), 00037950
2915* 7(HGAUSS(1, 7), U(1)),(HGAUSS(1, 8), V(1)),(HGAUSS(1, 9), W(1)), 00037960
2916* 8(HGAUSS(1,10), X(1)),(HGAUSS(1,11), Y(1)),(HGAUSS(1,12), Z(1)), 00037970
2917* 9(HGAUSS(1,13), AA(1)),(HGAUSS(1,14), BB(1)),(HGAUSS(1,15), CC(1)), 00037980
2918* T(HGAUSS(1,16), DD(1)),(BZEROS(1, 1), EE(1)),(BZEROS(120,1), FF(1)), 00037990
2919* 1(BZEROS(1, 2), GG(1)),(BZEROS(120,2), HH(1)) 00038000
2920* DATA A,B,C,D,E,F,G,H,I,J,K,L,M 00038010
2921* N/- .4472136, 0.4472136, -0.6546537, 0.0000000, 0.6546537, -0.7650554, 0.0038020
2922* 1-0.2852315, 0.2852315, 0.7650553, -0.8302239, -0.4688488, 0.0000000, 0.0038030
2923* 2 0.4688488, 0.8302239, -0.8717402, -0.5917003, -0.2092993, 0.2092991, 0.0038040
2924* 3 0.5917001, 0.8717400, -0.8997580, -0.6771863, -0.3631175, 0.0000000, 0.0038050
2925* 4 0.3631175, 0.6771863, 0.8997580, -0.9195339, -0.7387739, -0.4779250, 0.0038060
2926* 5-0.1652790, 0.1652789, 0.4779249, 0.7387738, 0.9195338, -0.9340014, 0.0038070
2927* 6-0.7844836, -0.5652354, -0.2957582, 0.0000000, 0.2957581, 0.5652353, 0.0038080
2928* 7 0.7844834, 0.9340014, -0.9448975, -0.8192815, -0.6328754, -0.3995310, 0.0038090
2929* 8-0.1365529, 0.1365529, 0.3995309, 0.6328753, 0.8192813, 0.9448975, 0.0038100
2930* 9-0.9533069, -0.8463538, -0.6861843, -0.4829108, -0.2492869, 0.0000000, 0.0038110
2931* T 0.2492868, 0.4829106, 0.6861842, 0.8463537, 0.9533068, -0.9599299, 0.0038120
2932* 1-0.8678104, -0.7288621, -0.5506417, -0.3427235, -0.1163319, 0.1163318, 0.0038130
2933* 2 0.3427235, 0.5506415, 0.7288620, 0.8678104, 0.9599298, -0.9652544, 0.0038140
2934* 3-0.8850636, -0.7635341, -0.6062477, -0.4206389, -0.2153539, 0.0000000, 0.0038150
2935* 4 0.2153538, 0.4206389, 0.6062477, 0.7635341, 0.8850635, 0.9652544, 0.0038160
2936* 5-0.9695861, -0.8991729, -0.7920153, -0.6511131, -0.4860575, -0.2998304, 0.0038170
2937* 6-0.1013263, 0.1013262, 0.2998304, 0.4860575, 0.6523930, 0.7920151, 0.0038180
2938* 7 0.8991728, 0.9695860/ 00038190
2939* DATA N,O 00038200
2940* N/- .9731405, -0.9108602, -0.8157166, -0.6910172, -0.5413883, -0.3721744, 0.0038210
2941* 1-0.1895120, 0.0000000, 0.1895119, 0.3721744, 0.5413882, 0.6910170, 0.0038220
2942* 2 0.8157164, 0.9108602, 0.9731404, -0.9602899, -0.7966665, -0.5255324, 0.0038230
2943* 3-0.1834346, 0.1834346, 0.5255324, 0.7966665, 0.9602899/ 00038240
2944* DATA P,Q,R,S,T,U,V,W,X,Y,Z,AA,BB 00038250
2945* N/O .8333334, 0.8333331, 0.5444443, 0.7111111, 0.5444444, 0.3784749, 0.0038260
2946* 1 0.5548583, 0.5548581, 0.3784750, 0.2768261, 0.4317453, 0.4876190, 0.0038270
2947* 2 0.4317455, 0.2768261, 0.2107044, 0.3411230, 0.4124591, 0.4124591, 0.0038280
2948* 3 0.3411230, 0.2107046, 0.1654953, 0.2745391, 0.3464290, 0.3715193, 0.0038290
2949* 4 0.3464290, 0.2745388, 0.1654955, 0.1333061, 0.2248897, 0.2920431, 0.0038300
2950* 5 0.3275404, 0.3275403, 0.2920429, 0.2248897, 0.1333061, 0.1096126, 0.0038310
2951* 6 0.1871701, 0.2480485, 0.2868792, 0.3002176, 0.2868798, 0.2480485, 0.0038320
2952* 7 0.1871700, 0.1096126, 0.0916847, 0.1579750, 0.2125089, 0.2512758, 0.0038330
2953* 8 0.2714060, 0.2714060, 0.2512759, 0.2125089, 0.1579748, 0.0916846, 0.0038340
2954* 9 0.0778019, 0.1349820, 0.1836473, 0.2207679, 0.2440163, 0.2519308, 0.0038350
2955* T 0.2440165, 0.2207679, 0.1836473, 0.1349820, 0.0778019, 0.0668373, 0.0038360
2956* 1 0.1165870, 0.1600221, 0.1948268, 0.2191266, 0.2316136, 0.2316138, 0.0038370
2957* 2 0.2191266, 0.1948268, 0.1600223, 0.1165869, 0.0668375, 0.0580301, 0.0038380
2958* 3 0.1016605, 0.1405119, 0.1727902, 0.1969877, 0.2119743, 0.2170480, 0.0038390
2959* 4 0.2119743, 0.1969876, 0.1727903, 0.1405120, 0.1016601, 0.0580301, 0.0038400
2960* 5 0.0508505, 0.0893939, 0.1242559, 0.1540275, 0.1774924, 0.1936907, 0.0038410
2961* 6 0.2019594, 0.2019594, 0.1936906, 0.1774925, 0.1540275, 0.1242554, 0.0038420
2962* 7 0.0893940, 0.0508506/ 00038430
2963* DATA CC,DD 00038440
2964* N/O .0449221, 0.0791985, 0.1105931, 0.1379879, 0.1603954, 0.1770052, 0.0038450
2965* 1 0.1872171, 0.1906618, 0.1872172, 0.1770049, 0.1603951, 0.1379883, 0.0038460
2966* 2 0.1105928, 0.0791985, 0.0449221, 0.1012285, 0.2223810, 0.3137067, 0.0038470

```


2967*	3 0.3626838, 0.3626838, 0.3137067, 0.2223810, 0.1012285/	00038480
2968*	DATA EE /2.404826, 5.520078, 8.653728, 11.79153, 14.93092,	00038490
2969*	1 18.07106, 21.21164, 24.35247, 27.49348, 30.63461, 33.77582,	00038500
2970*	2 36.91710, 40.05843, 43.19979, 46.34119, 49.48261, 52.62405,	00038510
2971*	3 55.76551, 58.90698, 62.04847, 65.18996, 68.33147, 71.47298,	00038520
2972*	4 74.61450, 77.75603, 80.89756, 84.03909, 87.18063, 90.32217,	00038530
2973*	5 93.46372, 96.60527, 99.74682, 102.8883, 106.0299, 109.1715,	00038540
2974*	6 112.3131, 115.4546, 118.5962, 121.7377, 124.8793, 128.0209,	00038550
2975*	7 131.1624, 134.3040, 137.4456, 140.5872, 143.7287, 146.8703,	00038560
2976*	8 150.0119, 153.1535, 156.2950, 159.4366, 162.5782, 165.7198,	00038570
2977*	9 168.8613, 172.0029, 175.1445, 178.2861, 181.4277, 184.5692,	00038580
2978*	1 187.7108, 190.8524, 193.9940, 197.1356, 200.2772, 203.4187,	00038590
2979*	2 206.5603, 209.7019, 212.8435, 215.9850, 219.1267, 222.2682,	00038600
2980*	225.4098, 228.5514, 231.6930, 234.8346, 237.9762, 241.1178,	00038610
2981*	3 244.2593, 247.4009, 250.5425, 253.6841, 256.8257, 259.9673,	00038620
2982*	4 263.1089, 266.2504, 269.3920, 272.5336, 275.6752, 278.8168,	00038630
2983*	5 281.9584, 285.1000, 288.2416, 291.3831, 294.5247, 297.6663,	00038640
2984*	6 300.8079, 303.9495, 307.0911, 310.2327, 313.3743, 316.5159,	00038650
2985*	7 319.6574, 322.7990, 325.9406, 329.0822, 332.2238, 335.3654,	00038660
2986*	8 338.5070, 341.6486, 344.7902, 347.9317, 351.0733, 354.2149,	00038670
2987*	9 357.3565, 360.4981, 363.6397, 366.7813, 369.9229, 373.0645,	00038680
2988*	DATA FF /376.2061, 379.3476, 382.4892, 385.6308, 388.7724,	00038690
2989*	1 391.9140, 395.0556, 398.1972, 401.3388, 404.4804, 407.6220,	00038700
2990*	2 410.7635, 413.9051, 417.0467, 420.1883, 423.3299, 426.4715,	00038710
2991*	3 429.6131, 432.7547, 435.8963, 439.0379, 442.1794, 445.3210,	00038720
2992*	4 448.4626, 451.6042, 454.7458, 457.8874, 461.0290, 464.1706,	00038730
2993*	5 467.3122/	00038740
2994*	DATA GG /3.831706, 7.015587, 10.17347, 13.32369, 16.47063,	00038750
2995*	1 19.61586, 22.76008, 25.90367, 29.04683, 32.18968, 35.33231,	00038760
2996*	2 38.47477, 41.61709, 44.75932, 47.90146, 51.04354, 54.18555,	00038770
2997*	3 57.32753, 60.46946, 63.61136, 66.75323, 69.89507, 73.03690,	00038780
2998*	4 76.17870, 79.32049, 82.46226, 85.60402, 88.74577, 91.88750,	00038790
2999*	5 95.02923, 98.17095, 101.3127, 104.454, 107.5961, 110.7376,	00038800
3000*	6 113.8794, 117.0211, 120.1628, 123.3045, 126.4461, 129.5878,	00038810
3001*	7 132.7295, 135.8711, 139.0128, 142.1544, 145.2961, 148.4377,	00038820
3002*	8 151.5794, 154.7210, 157.8626, 161.0043, 164.1459, 167.2876,	00038830
3003*	9 170.4292, 173.5708, 176.7125, 179.8541, 182.9957, 186.1374,	00038840
3004*	1 189.2790, 192.4206, 195.5622, 198.7038, 201.8455, 204.9871,	00038850
3005*	2 208.1287, 211.2703, 214.4120, 217.5536, 220.6952, 223.8368,	00038860
3006*	3 226.9784, 230.1200, 233.2616, 236.4033, 239.5449, 242.6865,	00038870
3007*	4 245.8281, 248.9697, 252.1113, 255.2529, 258.3945, 261.5362,	00038880
3008*	5 264.6778, 267.8194, 270.9610, 274.1026, 277.2442, 280.3858,	00038890
3009*	6 283.5274, 286.6690, 289.8106, 292.9522, 296.0938, 299.2354,	00038900
3010*	7 302.3771, 305.5187, 308.6603, 311.8019, 314.9435, 318.0851,	00038910
3011*	8 321.2267, 324.3683, 327.5099, 330.6515, 333.7931, 336.9347,	00038920
3012*	9 340.0763, 343.2179, 346.3595, 349.5011, 352.6427, 355.7843,	00038930
3013*	1 358.9259, 362.0675, 365.2091, 368.3507, 371.4923, 374.6339,	00038940
3014*	DATA HH /377.7755, 380.9171, 384.0587, 387.2003, 390.3419,	00038950
3015*	2 393.4835, 396.6251, 399.7667, 402.9083, 406.0499, 409.1919,	00038960
3016*	3 412.3331, 415.4747, 418.6163, 421.7579, 424.8995, 428.0411,	00038970
3017*	4 431.1827, 434.3243, 437.4659, 440.6075, 443.7491, 446.8907,	00038980
3018*	5 450.0323, 453.1739, 456.3155, 459.4570, 462.5987, 465.7403,	00038990
3019*	6 468.8819/	00039000
3020*	END	00039010

ATE
MED
-8